



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

**MÁSTER UNIVERSITARIO EN TECNOLOGÍAS
INFORMÁTICAS AVANZADAS**

TRABAJO FIN DE MÁSTER

Algoritmo de pre-análisis para el estándar de
codificación de vídeo HEVC

Gabriel Cebrián Márquez

Julio de 2015



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

Departamento de Sistemas Informáticos

TRABAJO FIN DE MÁSTER

Algoritmo de pre-análisis para el estándar de
codificación de vídeo HEVC

Autor: D. Gabriel Cebrián Márquez

Directores: Dr. D. Pedro Ángel Cuenca Castillo

Dr. D. José Luis Martínez Martínez

Julio de 2015

Declaración de Autoría

Gabriel Cebrián Márquez, con DNI 47.096.875-M declara que este documento ha sido realizado exclusivamente por él durante el curso 2014/2015 y es fruto de su trabajo personal.

Toda la información utilizada de otros autores ha sido debidamente referenciada o citada, indicando explícitamente el autor último de los fragmentos utilizados.

Resumen

En los últimos años, y especialmente en el presente más actual, el consumo de contenidos multimedia ha crecido exponencialmente. Este hecho viene favorecido por el desarrollo de las redes de telecomunicaciones e Internet, y por el nacimiento de los llamados teléfonos inteligentes o *smartphones*. Este gran consumo hace necesario buscar formas de comprimir dichos contenidos de manera eficiente.

High Efficiency Video Coding (HEVC) es el último estándar de codificación elaborado por el *Joint Collaborative Team on Video Coding* (JCT-VC), el cual es capaz de doblar la tasa de compresión de una secuencia de vídeo alcanzando la misma calidad que el estándar inmediatamente anterior, H.264/AVC. Teniendo en cuenta estas prestaciones, las empresas han mostrado un gran interés en él. No obstante, este incremento de la eficiencia de codificación es conseguido mediante el uso de técnicas mejoradas y nuevas estrategias de compresión que aumentan la complejidad computacional del *encoder* en varios órdenes de magnitud.

Pese a que el estándar ha definido un conjunto de técnicas de paralelización que es posible utilizar en el proceso de codificación, sufren de una serie de limitaciones que obligan a buscar alternativas para reducir todavía más el tiempo de codificación. Así, resulta necesario buscar y desarrollar técnicas que permitan reducir la complejidad del *encoder*.

Este documento describe la arquitectura de un algoritmo de pre-análisis para un codificador basado en el estándar HEVC. Este algoritmo permite obtener información previa al proceso de codificación que puede ser utilizada en los diversos módulos que componen al *encoder* y así reducir la carga de trabajo que han de llevar a cabo. Al tratarse de estimaciones, su cálculo supone poco coste computacional, ayudando a alcanzar una notable reducción del tiempo de codificación, mientras que la eficiencia de codificación apenas se ve afectada.

En este sentido, se mostrará el diseño, la parametrización y los resultados de la parte del algoritmo de pre-análisis que da soporte a la estimación de movimiento. Esta parte obtiene una aproximación de los vectores de movimiento que componen la imagen a codificar, pudiendo ser empleados como punto de partida en el módulo de estimación de movimiento. Los resultados muestran que gracias a este algoritmo, es posible reducir el tiempo de codificación en un 15 % con tan sólo un 0.9 % de incremento en BD-rate.

Abstract

In recent years, the consumption of multimedia contents has grown exponentially. This fact is given by the development of the communication networks and the Internet, and also by the birth of the so-called smartphones. This enormous consumption makes it necessary to search for ways of compressing these contents in an efficient way.

High Efficiency Video Coding (HEVC) is the latest coding standard developed by the *Joint Collaborative Team on Video Coding* (JCT-VC). This standard is able to double the compression rate of a video sequence compared to the previous standard, H.264/AVC, while maintaining the same objective quality. This is the main feature that has attracted a lot of attention from many commercial companies. Nonetheless, this coding efficiency improvement is achieved by the use of enhanced techniques and new coding strategies that increase the computational complexity of the encoder by several orders of magnitude.

Even though the standard defines a set of parallelization techniques that can be used in the encoding process, they suffer from some limitations that lead us to search for alternatives in order to reduce the encoding time even further. In this way, it is necessary to develop techniques that reduce the complexity of the encoder.

This document describes the architecture of a HEVC-compliant pre-analysis algorithm. This algorithm allows to obtain information prior the encoding process, which can be used in the modules that the encoder comprises so as to reduce the processing time. Given that it is an estimation of the real data, its calculation involves a low computational cost. Nevertheless, it allows to achieve a notable reduction of the computational complexity of the encoder, whereas the coding efficiency is negligibly affected.

As a part of the whole algorithm, the design, parametrisation and results of the motion estimation pre-analysis algorithm will also be shown. This part is able to obtain an approximation of the motion vectors of a picture, so that they can be used as starting point in the motion estimation module of the encoder. Results show that the proposed algorithm achieves a time reduction of 15 % of the encoding process with only a 0.9 % increment of BD-rate.

*A todas las personas que amo
y, en especial, a aquéllas que ya no están.*

Agradecimientos

Un año más, he de agradecer este trabajo a todas aquellas personas que me rodean y que me han apoyado todo este tiempo. Este pequeño fragmento de texto, no obstante, no hace justicia a la enorme gratitud que quisiera transmitirles.

No sería posible comenzar de otra forma que agradeciendo a los tutores de este trabajo la oportunidad que me brindaron al contar conmigo en su gran proyecto, y también por el enorme esfuerzo que han realizado en todo momento. Gracias, José Luis y Pedro.

También querría agradecer al profesorado de la Escuela su excelente labor a la hora de transmitirnos sus conocimientos, pero especialmente por ser grandísimas personas y por enseñarnos aquellas cosas que no aparecen en los libros.

Igualmente necesario resulta agradecer el apoyo de todos los compañeros que forman el grupo de *Redes y Arquitecturas de Altas Prestaciones* (RAAP). Sin lugar a dudas, hacen las mañanas mucho más amenas y apacibles, siempre conservando la profesionalidad que les caracteriza.

Este pequeño párrafo no representa todo lo que me gustaría agradecer a mi familia. Los años pasan; vosotros permanecéis ahí. Chache, a pesar de tu fuerte temperamento, eres una gran persona que sabe apoyar a las personas que quiere. Te deseo lo mejor en tu aventura por Estados Unidos. Mamá, no podríamos ser más afortunados de tenerte. No he conocido a otra persona que se desviva por sus hijos como tú lo haces por nosotros. Papá, más de tres años han pasado desde que la infame enfermedad de cáncer se llevase consigo a una de las personas más valientes, honestas y queridas por su familia y amigos. Espero que te sientas orgulloso de nosotros y que algún día me convierta en el padre que tú has sido para nosotros.

Reservo las últimas palabras de estos agradecimientos para ti, Estefanía. Hemos compartido casi la quinta parte de nuestras vidas, pero me ha sido necesario mucho menos que eso para descubrir la gran persona que hay en tu interior. No puedo expresar lo afortunado que me siento de tenerte a mi lado. Nuestra pareja ha pasado por muchos retos y estoy seguro de que se mantendrá fuerte frente a aquéllos que están por venir.

A todos vosotros, mi más sincero cariño y gratitud... ¡gracias!

Este trabajo ha sido cofinanciado por el Ministerio de Economía y Competitividad y la Comisión Europea (fondos FEDER) bajo el proyecto TIN2012-38341-C04-04, y por el Ministerio de Educación, Cultura y Deporte mediante la beca FPU 13/04601.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABLAS	XV
LISTA DE ACRÓNIMOS	XVII
1. INTRODUCCIÓN	1
1.1. Motivación	2
1.2. Objetivos	3
1.3. Metodología	4
1.4. Estructura de la memoria	4
2. ASIGNATURAS CURSADAS	7
2.1. Metodologías y Técnicas de Investigación en Informática	7
2.2. Redes Multimedia	8
2.3. Introducción a la Programación de Arquitecturas de Altas Prestaciones	9
2.4. Sistemas Inteligentes Aplicados a Internet	11
2.5. Modelado y Evaluación de Sistemas	12
2.6. Modelos para el Análisis y Diseño de Sistemas Concurrentes	13
2.7. Grid Computing	15
3. FUNDAMENTOS DEL ESTÁNDAR HEVC	17
3.1. Consideraciones de diseño	18
3.2. División de un frame en unidades básicas	19
3.3. División de un frame en unidades mayores	21
3.4. Predicción intra	23
3.5. Predicción inter	24
3.6. Transformada, cuantización y entropía	27
3.7. Filtros in-loop	28
3.8. Técnicas de paralelización	29
3.8.1. <i>Tiles</i>	30
3.8.2. <i>Wavefront Parallel Processing</i> (WPP)	31
3.8.3. Aplicaciones de cada estrategia de paralelización	31
3.9. Perfiles, niveles y tiers	32
3.10. Patrones de codificación	33

4. TRABAJO DE INVESTIGACIÓN: ALGORITMO DE PRE-ANÁLISIS PARA HEVC	35
4.1. Motivación	36
4.2. Trabajos relacionados	38
4.3. Algoritmo de pre-análisis para la ME	41
4.3.1. Etapa de pre-análisis	42
4.3.2. Etapa integrada en la ME	43
4.4. Configuración del algoritmo de pre-análisis para la ME	45
4.4.1. Métricas empleadas	45
4.4.2. Configuración experimental	46
4.4.3. Tamaño del bloque de pre-análisis	48
4.4.4. Rango de búsqueda del módulo ME	49
4.4.5. Sustitución del algoritmo de búsqueda en el módulo ME	52
4.5. Resumen de resultados	56
4.6. Conclusiones	58
5. ANTEPROYECTO DE TESIS DOCTORAL	59
5.1. Contribuciones a la Tesis Doctoral	59
5.2. Descripción del anteproyecto de Tesis Doctoral	60
5.2.1. Introducción	60
5.2.2. Objetivos	61
5.2.3. Tareas a realizar	62
5.2.4. Cronograma temporal	64
5.2.5. Recursos necesarios	66
BIBLIOGRAFIA	70
A. CURRICULUM VITAE	71

ÍNDICE DE FIGURAS

3.1. Línea temporal de los estándares de vídeo	18
3.2. Diagrama de bloques de un codificador basado en HEVC	19
3.3. División de un CTU en CUs siguiendo una estructura de <i>quadtree</i> . . .	20
3.4. Esquemas de particionamiento de un CU en PUs	20
3.5. Fragmento en detalle de la partición de un <i>frame</i> en CUs y TUs	22
3.6. División de un <i>frame</i> en <i>slices</i>	22
3.7. Modos y orientaciones para la predicción intra	24
3.8. Posiciones enteras y sub-píxel en el filtro de interpolación	25
3.9. Posiciones de los candidatos espaciales del modo <i>Merge</i>	26
3.10. Escalado del predictor de movimiento de un PU vecino en otro <i>frame</i> .	27
3.11. Particionamiento de un <i>frame</i> en <i>tiles</i> y su correspondiente orden de procesamiento	30
3.12. Procesamiento en filas de WPP	31
3.13. Patrones de codificación <i>Low Delay</i> y <i>Random Access</i> , y sus referencias	33
4.1. Perfil de tiempo para la secuencia Kimono, <i>Random Access</i> , QP32 . . .	36
4.2. Diagrama de bloques de un codificador basado en HEVC empleando pre-análisis	38
4.3. Ejemplo de funcionamiento de la estrategia OWF con cuatro hilos . . .	39
4.4. Diagrama de bloques de un codificador basado en HEVC empleando pre-análisis (en rojo)	42
4.5. Soporte de la etapa de pre-análisis en el algoritmo de elección de predictores	44
4.6. Ejemplo del cálculo de BD-rate sobre curvas de bit-rate/PSNR	46
4.7. Resultados de BD-rate para diferentes tamaño de bloque de pre-análisis	49
4.8. Resultados de BD-rate para diferentes rangos de búsqueda sin el algo- ritmo de pre-análisis	50
4.9. Resultados de BD-rate y tiempo para diferentes rangos de búsqueda del módulo ME aplicando el algoritmo de pre-análisis	52
4.10. Pasos ejecutados en el algoritmo de búsqueda por defecto	53
4.11. Algoritmo de búsqueda propuesto con patrón hexagonal	53
4.12. Resultados de BD-rate y tiempo para diferentes rangos de búsqueda aplicando el algoritmo de pre-análisis y el nuevo algoritmo de búsqueda	55
4.13. Comparación entre el codificador base y el codificador con el algoritmo de pre-análisis	57

5.1. Planificación temporal de la Tesis Doctoral propuesta 65

ÍNDICE DE TABLAS

3.1. Coeficientes aplicados en los filtros <i>eight-tap</i> y <i>seven-tap</i> de interpolación sub-píxel	25
3.2. Comparación de las estrategias de paralelización disponibles en HEVC .	31
4.1. Resultados de BD-rate para diferentes tamaño de bloque de pre-análisis	48
4.2. Resultados de BD-rate para diferentes rangos de búsqueda del módulo ME aplicando el algoritmo de pre-análisis	51
4.3. Resultados de tiempo para diferentes rangos de búsqueda del módulo ME aplicando el algoritmo de pre-análisis	51
4.4. Resultados de BD-rate para diferentes rangos de búsqueda aplicando el algoritmo de pre-análisis y el nuevo algoritmo de búsqueda	54
4.5. Resultados de tiempo para diferentes rangos de búsqueda aplicando el algoritmo de pre-análisis y el nuevo algoritmo de búsqueda	55
4.6. Resultados del algoritmo de pre-análisis empleando la configuración óptima propuesta	57

LISTA DE ACRÓNIMOS

AMP	Asymmetric Motion Partitioning
AMVP	Advanced Motion Vector Prediction
AVC	Advanced Video Coding
BLAS	Basic Linear Algebra Subprograms
CABAC	Context-Adaptive Binary Arithmetic Coding
CCS	Calculus of Communicating Systems
CSP	Communicating Sequential Processes
CTU	Coding Tree Unit
CU	Coding Unit
DBF	Deblocking Filter
DCT	Discrete Cosine Transform
DPB	Decoded Picture Buffer
DST	Discrete Sine Transform
FME	Fractional Motion Estimation
FPGA	Field-Programmable Gate Array
FPS	Frames Per Second
GOP	Group Of Pictures
GPU	Graphics Processing Unit
HEVC	High Efficiency Video Coding
IDCT	Inverse Discrete Cosine Transform
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers

IME	Integer Motion Estimation
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union-Telecommunication
ITU-R	International Telecommunication Union Radiocommunications Sector
JCR	Journal Citation Reports
JCT-VC	Joint Collaborative Team on Video Coding
MC	Motion Compensation
ME	Motion Estimation
MER	Motion Estimation Region
MPEG	Moving Picture Experts Group
MPI	Message Passing Interface
MPM	Most Probable Mode
MSE	Mean Squared Error
MV	Motion Vector
MVP	Motion Vector Predictor
OWF	Overlapped Wavefront
PSNR	Peak Signal to Noise Ratio
PU	Prediction Unit
RC	Rate Control
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
ROI	Region Of Interest
ROSA	Reasoning On Stochastic Algebras
RQT	Residual Quadtree
QP	Quantization Parameter
SAD	Sum of Absolute Differences

SAO	Sample Adaptive Offset
ScaLAPACK	Scalable Linear Algebra Package
SIMD	Single Instruction Multiple Data
TU	Transform Unit
UHD	Ultra High Definition
VCEG	Video Coding Experts Group
VHS	Video Home System
WAN	Wide Area Network
WiMAX	Worldwide Interoperability for Microwave Access
WPP	Wavefront Parallel Processing

Capítulo 1

INTRODUCCIÓN

Hace algo más de un par de décadas, la mayor parte de nuestro entorno era analógico, tal y como demuestra el uso de discos de vinilo o cintas de vídeo VHS. No obstante, a raíz de las grandes limitaciones que este formato supone en términos de almacenamiento y transmisión de contenidos, surgió una necesidad imperiosa por desplazarlo en favor del formato digital. Entre otras ventajas, los contenidos digitales tienen una mayor capacidad de conservación en el tiempo, pueden beneficiarse de las posibilidades que ofrecen los algoritmos de compresión y son más fácilmente transmisibles por las redes de comunicaciones.

En este punto es donde debemos señalar la importancia de los codificadores de vídeo (o *video encoders*), los cuales se encargan de comprimir y/o codificar una señal de vídeo de entrada en un flujo de bits que puede ser almacenado y transmitido en formato digital. Un decodificador (o *video decoder*), por su parte, se encarga de transformar las cadenas de bits recibidas en los distintos cuadros que forman una secuencia de vídeo.

Gracias al formato de vídeo digital, el consumo de contenidos multimedia no ha parado de crecer en los últimos años, al igual que la calidad exigida a los mismos, llegando a resoluciones tales como 4k (*Ultra High Definition*, UHD) o incluso 8k. Desafortunadamente, esto genera una serie de problemas relacionados que debemos resolver:

- Los contenidos generados son cada vez mayores en número y de mayor calidad, siendo necesario, por tanto, una gran cantidad de bits para representar los flujos de vídeo originales.
- Las redes de comunicaciones, como puede ser Internet, no evolucionan a la misma velocidad que las exigencias de los usuarios. De hecho, Cisco [1] prevé que el tráfico correspondiente a vídeo del tráfico total en Internet ascenderá del 64 % en 2014 al 80 % en 2019.

Se hace notable, por tanto, la importancia del codificador de vídeo, el cual se debe encargar de realizar una buena compresión de este tipo de contenidos manteniendo

una calidad aceptable. A este objetivo responden los diversos estándares de vídeo que han sido elaborados a lo largo del tiempo: MPEG-1 [2], MPEG-2 [3], MPEG-4 Visual [4], H.264/AVC [5]... y que indican la sintaxis del flujo codificado para permitir interoperabilidad entre dispositivos.

Sin embargo, los codificadores de vídeo son cada vez más exigentes en términos de coste computacional, llegando a ser impracticable en algunos casos. Este hecho nos obliga a buscar técnicas que nos permitan obtener el flujo de bits resultante en el menor tiempo posible. El reto propuesto para este Trabajo Fin de Máster, por tanto, será el de desarrollar algoritmos que permitan acelerar el proceso de codificación manteniendo la eficiencia de codificación.

1.1. Motivación

El estándar *High Efficiency Video Coding* (HEVC) [6] es un estándar de compresión de vídeo que pretende ser el sucesor directo del omnipresente H.264 *Advanced Video Coding* (H.264/AVC) [5]. Su desarrollo fue finalizado a inicios del año 2013, si bien se han publicado diversas extensiones desde entonces con el objetivo de dar soporte a escenarios adicionales.

La aparición de este estándar vino motivada por la necesidad de mejorar la eficiencia de compresión de su antecesor, el estándar H.264/AVC, especialmente acentuada con la llegada de resoluciones en torno al 4k y 8k, los modos multivista y el excesivo tráfico de vídeo en la red. Muchas empresas tienen un gran interés en él debido a las capacidades que ha logrado mostrar hasta el momento [7]: un 50 % de reducción en tasa de bits bajo la misma calidad de vídeo o, visto de otro modo, una mejora en la calidad de vídeo con un flujo de igual caudal de bits.

Esta mejora en compresión, sin embargo, lleva asociada una complejidad computacional varios órdenes de magnitud mayor comparado con estándares anteriores [7]. Por ejemplo, aunque resulta difícil comparar dos codificadores de estándares distintos, el *encoder* de referencia para H.264/AVC, JM [8], codificaba una secuencia de vídeo entre 3 y 4 veces más rápido que el codificador de referencia de HEVC, HM [9], en condiciones similares de configuración [10].

Afortunadamente, este coste en tiempo puede ser reducido en buena parte mediante el uso de técnicas de paralelización. En este sentido, el propio estándar HEVC define sus propias técnicas, las cuales serán descritas en la Sección 3.8. Aun así, esto no es suficiente para reducir el tiempo de codificación a unos niveles razonables. Es por ello que resulta necesario desarrollar algoritmos que permitan reducir la complejidad computacional del codificador, bien de manera aislada o bien en colaboración con los algoritmos paralelos.

En este escenario, este Trabajo Fin de Máster surge a raíz de la estancia llevada a cabo durante tres meses en el grupo *Embedded Systems Architecture* (AES) de la *Technische Universität Berlin* [11]. En colaboración con algunos de sus miembros, se concibió la idea de un algoritmo de pre-análisis que diera soporte al proceso de codificación, de manera que el tiempo total de codificación se viera reducido en su conjunto. Así, después de un extenso análisis del estándar y otros trabajos, se llevó a cabo un algoritmo de pre-análisis para el módulo de estimación de movimiento del codificador de HEVC.

Adicionalmente, el trabajo descrito en este documento representa la base de la Tesis Doctoral a desarrollar. Por este motivo, además de detallar los aspectos más importantes de dicho algoritmo y las decisiones de diseño que le precedieron, esta memoria también busca mostrar las múltiples líneas de trabajo futuras que surgen a raíz de él, y que serán cubiertas a lo largo del desarrollo de la Tesis.

1.2. Objetivos

Teniendo en cuenta la finalidad de este trabajo mencionada al final del apartado anterior, los objetivos principales a desarrollar se resumen en los siguientes puntos:

- Relacionar las asignaturas cursadas en el Máster Universitario de Tecnologías Informáticas Avanzadas con la línea de trabajo del presente documento.
- Realizar un estudio de las características del estándar HEVC, incluyendo los módulos que típicamente forman parte de un codificador basado en éste. De esta manera, se podrá analizar cuáles son las partes más interesantes a cubrir por el algoritmo de pre-análisis propuesto.
- Proponer la arquitectura del algoritmo de pre-análisis que se llevará a cabo a lo largo de la Tesis Doctoral.
- Introducir los detalles de la parte del algoritmo desarrollada para este Trabajo Fin de Máster, la cual surge a raíz del estudio anterior y tiene como objetivo dar soporte a la estimación de movimiento del codificador.
- Razonar y evaluar cada una de las decisiones de diseño que han dado lugar al algoritmo propuesto.
- Mostrar las futuras líneas de trabajo de la Tesis Doctoral a la que dará pie este trabajo de investigación, así como su planificación.

1.3. Metodología

La metodología seguida en el desarrollo de este trabajo está dividida en tres grandes partes: una primera fase de análisis, una segunda de desarrollo y la última fase correspondiente a la parametrización y evaluación del algoritmo.

1. La fase de análisis, a su vez, se divide en tres subtarefas:

- a) En primer lugar, llevar a cabo una lectura en detalle del estándar HEVC. A partir de ello, se tendrá un conocimiento profundo de sus características que servirá como base para pasos posteriores.
- b) Buscar y analizar otras propuestas existentes en esta línea de trabajo. Será necesario, además, examinar las ventajas y desventajas de cada una de ellas, si procede.
- c) Realizar un *profiling* sobre el codificador en que se implementará la propuesta para obtener los tiempos correspondientes a cada uno de los módulos que lo componen. Esto dará idea de cuáles de ellos son los que más tiempo consumen en el proceso de codificación, y que se convertirán en el foco de atención del algoritmo propuesto.

2. Durante la parte de desarrollo, primeramente es necesario decidir la arquitectura que tendrá nuestro algoritmo, así como de qué manera y con qué partes del codificador se encontrará relacionado. Una vez decidido, se podrá proceder a desarrollar el algoritmo que, en lo que concierne a este Trabajo Fin de Máster, se limitará al módulo de estimación de movimiento del codificador de HEVC.

3. Por último, la última fase constará de un proceso iterativo de parametrización, evaluación y mejora. En otras palabras, el algoritmo dispone de ciertos parámetros cuyos valores han de ser ajustados y evaluados para obtener el mayor rendimiento posible. Una vez establecidos, es posible que el algoritmo necesite pasar de nuevo a la fase de desarrollo para seguir siendo mejorado.

Una vez finalizadas todas las fases, será posible extraer una serie de conclusiones y las posibles líneas de desarrollo de la Tesis Doctoral.

1.4. Estructura de la memoria

A continuación, se detallarán los distintos capítulos de los que consta la presente memoria, indicando los contenidos de cada uno de ellos:

- Capítulo 1: INTRODUCCIÓN. Se trata del presente capítulo. En él hemos introducido brevemente el contexto de este Trabajo Fin de Máster, así como su alcance y los objetivos a cumplir.
- Capítulo 2: ASIGNATURAS CURSADAS. Este capítulo presenta las asignaturas que se han cursado a lo largo del Máster Universitario en Tecnologías Informáticas Avanzadas. Se describirá su contenido y los trabajos realizados en cada una de ellas, así como su relación con la Tesis Doctoral descrita en el documento.
- Capítulo 3: FUNDAMENTOS DEL ESTÁNDAR HEVC. Este capítulo tratará algunos de los aspectos más importantes del estándar de codificación de vídeo HEVC.
- Capítulo 4: TRABAJO DE INVESTIGACIÓN: ALGORITMO DE PRE-ANÁLISIS PARA HEVC. En él se encuentra descrito el algoritmo de pre-análisis propuesto en este Trabajo Fin de Máster, así como todas las fases de diseño que han dado lugar al algoritmo de soporte a la estimación de movimiento desarrollado. Al final del mismo, se incluyen los resultados obtenidos.
- Capítulo 5: ANTEPROYECTO DE TESIS DOCTORAL. Este capítulo contiene la descripción de los siguientes pasos que se llevarán a cabo en la Tesis Doctoral a la cual da pie este trabajo de investigación, así como su planificación.

Capítulo 2

ASIGNATURAS CURSADAS

En el presente capítulo, se hará una breve descripción de las asignaturas cursadas dentro del Máster Universitario en Tecnologías Informáticas Avanzadas. Esta descripción incluye los contenidos impartidos, los trabajos realizados y la relación de la asignatura con la Tesis Doctoral. Todas las asignaturas tienen una carga lectiva de cinco créditos ECTS y han sido cursadas en el Campus de Albacete, a excepción de la asignatura Metodologías y Técnicas de Investigación en Informática que fue impartida desde Ciudad Real.

2.1. Metodologías y Técnicas de Investigación en Informática

Esta asignatura constituye una de las bases de la investigación. Es impartida por Dr. D. Mario Piattini Velthuis, Dr. D. José Antonio Cruz Lemus y Dra. D.^a Marcela Genero Bocco, todos ellos integrantes del grupo Alarcos, centrado en la calidad y sostenibilidad de los sistemas de información.

La asignatura se marca como objetivo principal presentar los métodos de investigación más adecuados para la validación y contrastación de las hipótesis de investigación, así como ofrecer a los alumnos el conjunto de técnicas necesario para realizar una Tesis Doctoral.

Entre las técnicas descritas a lo largo de la asignatura, la más importante es la realización de una revisión sistemática de la literatura. Mediante esta técnica, es posible realizar un análisis exhaustivo del estado de arte que precede a la realización de una Tesis Doctoral. Este análisis viene marcado por objetivos y preguntas que han de responderse a través de los trabajos ya realizados en la temática.

Trabajo realizado en la asignatura

Como trabajo de evaluación de la asignatura, se realizó una revisión sistemática de la literatura relacionada con la línea de Tesis Doctoral presentada en este documento. En el caso que nos ocupa, se llevó a cabo un estudio del estado de arte relacionado con HEVC y, más específicamente, de aquellos trabajos publicados hasta el momento que hicieran uso de arquitecturas heterogéneas para el proceso de codificación, como por ejemplo mediante *Graphics Processing Units* (GPUs) o *Field-Programmable Gate Arrays* (FPGAs). En este sentido, la intención de este trabajo no es sólo conocer el estado de arte en sí, sino aplicar la metodología impartida durante la asignatura para llevar a cabo el estudio. La evaluación de dicho trabajo constó de una memoria escrita y una exposición oral.

Relación de la asignatura con la Tesis Doctoral

Esta asignatura provee las técnicas y la metodología necesarias para hacer una revisión de la literatura que, si bien está enfocada al inicio de una Tesis Doctoral, puede ser aplicada de manera constante a lo largo de todo su desarrollo. En este sentido, resulta enormemente útil durante toda la carrera investigadora del alumno, especialmente para conocer las líneas de investigación existentes en una temática y los nichos donde es necesario un mayor esfuerzo.

2.2. Redes Multimedia

Esta asignatura, la cual comprende un amplio espectro de temáticas pertenecientes a redes y a contenidos multimedia, es impartida por diversos miembros del grupo de *Redes y Arquitecturas de Altas Prestaciones* (RAAP).

En primer lugar, Dr. D. Antonio Garrido del Solo introdujo los principales conceptos relacionados con la calidad de servicio en redes cableadas, así como los posibles problemas que pueden darse en éstas y sus soluciones. Posteriormente, Dr. D. José Miguel Villalón Millán trasladó esta temática a las redes inalámbricas de tipo IEEE 802.11, especialmente a las transmisiones de tipo multicast. Tras ello, Dr. D. Francisco Manuel Delicado Martínez presentó tecnologías destinadas a proveer acceso inalámbrico a Internet en redes de área amplia (*Wide Area Network*, WAN), como por ejemplo *Worldwide Interoperability for Microwave Access* (WiMAX), así como las posibilidades de calidad de servicio que ofrecen.

Alejándose un poco de las redes de comunicación entre computadores, Dra. D.^a Teresa Olivares Montes centró su bloque de contenidos en la temática de redes de

sensores, donde fueron presentados sus principales características, sus aplicaciones y el estándar IEEE 802.15.4.

Por otro lado, los profesores Dr. D. José Luis Martínez Martínez y Dr. D. Pedro Ángel Cuenca Castillo introdujeron los detalles de los principales estándares de codificación de vídeo empleados en la actualidad, los cuales se corresponden con H.264/AVC y HEVC junto con sus correspondientes extensiones. Finalmente, Dr. D. Gerardo Fernández Escribano partió de los contenidos explicados con anterioridad para aplicarlos a la transcodificación entre estándares de contenidos multimedia.

Trabajo realizado en la asignatura

Si bien se llevaron a cabo diversas tareas a lo largo del curso, se realizó un trabajo de mayor envergadura al final de la asignatura que debía contemplar al menos una de las temáticas expuestas durante el mismo. Dada la afinidad con la línea de tesis, el trabajo realizado desarrolló las principales características del estándar HEVC e incluyó un análisis de algunos de los principales trabajos relacionados con el mismo.

Relación de la asignatura con la Tesis Doctoral

La parte de la asignatura correspondiente a multimedia se encuentra altamente relacionada con la línea de tesis propuesta, puesto que desarrolla los contenidos de codificación de vídeo más importantes, así como el propio estándar HEVC. Por ello, resulta de gran interés para adquirir los conocimientos técnicos que formarán parte de la base de la Tesis Doctoral.

2.3. Introducción a la Programación de Arquitecturas de Altas Prestaciones

Esta asignatura es impartida por Dr. D. Diego Cazorla López, Dr. D. Enrique Arias Antúnez y Dr. D. Juan José Pardo Mateo, pertenecientes todos ellos al grupo de *Sistemas Concurrentes y Tiempo Real* (RETICS). Su principal objetivo es dotar al alumno de técnicas que le permitan desarrollar códigos capaces de resolver problemas de una manera rápida y eficiente, especialmente mediante el uso de arquitecturas paralelas. Igualmente, proporciona la metodología requerida para evaluar las prestaciones de los algoritmos desarrollados.

En una primera parte, se enseñan algunas de las técnicas más importantes de optimización de aplicaciones secuenciales. Entre ellas, se encuentra el correcto uso de la

jerarquía de memoria y la caché mediante técnicas de programación en bloques, y la explotación de la localidad espacial y temporal de los datos.

En segundo lugar, se presenta *Message Passing Interface* (MPI) como paradigma de programación en sistemas de memoria distribuida. A través de una serie de prácticas y ejemplos, se analizan las ventajas y desventajas de utilizar esta filosofía frente a la de memoria compartida, como pueden ser la escalabilidad o el uso de la red de interconexión, respectivamente.

Trabajo realizado en la asignatura

La asignatura contempla tres prácticas que evalúan los conocimientos y desempeño del alumno en la temática. Estas prácticas suponen la resolución de diversos casos prácticos empleando las técnicas adquiridas y el modelo de programación MPI.

La primera de ellas puso de manifiesto la importancia de conocer la arquitectura del ordenador a la hora de desarrollar algoritmos óptimos. Así, se mostró la gran diferencia de rendimiento existente entre las distintas formas de recorrer estructuras de datos matriciales, llegando a ser hasta 8 ó 9 veces superior en algunos casos. Además, también se mostró el efecto positivo del empleo de bloques de datos en la complejidad computacional del algoritmo. Por último, se hizo uso de la librería *Basic Linear Algebra Subprograms* (BLAS) como ejemplo de algoritmo optimizado.

Las dos siguientes prácticas propusieron el uso de la librería de MPI para resolver los problemas de la primera desde un punto de vista distribuido. En una de ellas, el reparto de los datos se realizó de manera manual, donde además los resultados se compararon con aquéllos obtenidos en sistemas de memoria compartida. En la otra práctica, se presentó el conjunto de librerías *Scalable Linear Algebra Package* (ScaLAPACK) que permite hacerlo de manera más sencilla. Los resultados mostraron que, en términos de rendimiento, los sistemas de memoria compartida alcanzaron mejores tiempos para resolver el problema resuelto. No obstante, este paradigma es inviable para problemas en que el tamaño de los datos es muy elevada. Por su parte, los resultados también mostraron que el reparto manual de los datos ofreció mejores tiempos que el de la librería diseñada para ello.

Como broche final de la asignatura, se llevó a cabo un trabajo final en el que hubo que analizar qué tipo de paralelismo incluía el proyecto de tesis a desarrollar. En éste, se expuso cómo se empleó la GPU para acelerar la estimación de movimiento del codificador de referencia de HEVC, lo cual se desarrolló como Trabajo Fin de Grado y fue continuado posteriormente como Trabajo Fin de Máster del Máster Universitario en Ingeniería Informática.

Relación de la asignatura con la Tesis Doctoral

El estándar HEVC define una serie de algoritmos de paralelización que permiten reducir la complejidad computacional tanto del codificador como del decodificador. Estas técnicas se basan en el reparto de las particiones de la imagen actual entre varios elementos de procesamiento distintos, de tal manera que puedan procesarse concurrentemente. De esta manera, esta asignatura proporciona los conocimientos necesarios para poder llevar el codificador a una arquitectura de memoria distribuida, o incluso a aquéllas que hacen uso de MPI, como la de Intel Xeon Phi [12], y que podrían ser de utilidad en un futuro cuando se decida implementar el algoritmo propuesto en arquitecturas heterogéneas.

2.4. Sistemas Inteligentes Aplicados a Internet

La asignatura es impartida por Dr. D. Ismael García Varea, Dra. D.^a María Julia Flores Gallego y Dr. D. José Miguel Puerta Callejón pertenecientes al grupo de *Sistemas Inteligentes y Minería de Datos* (SIMD), y está dividida en tres partes que son descritas a continuación.

En una primera parte, se realiza una introducción al reconocimiento y clasificación de patrones a través de modelos basados en la Teoría de Bayes, o de otros como los clasificadores de los k -vecinos más cercanos, las funciones de distancia y las máquinas de vector soporte. Además, también se presentan distintos modelos paramétricos y no paramétricos para estimar las funciones correspondientes a densidades de población específicas.

El segundo bloque, por su parte, se centra en primer lugar en el diseño y modelado de redes bayesianas para, posteriormente, detallar el uso de métodos de inferencia con el objetivo de extraer información de las mismas.

La tercera y última parte de la asignatura establece diversos métodos y formas de estimar datos perdidos o modelos a partir de una serie existente de datos.

Trabajo realizado en la asignatura

La asignatura se encuentra planteada de tal manera que cada una de las tres partes mencionadas anteriormente incluye una serie de tareas a realizar.

En relación a la primera parte, se realizaron dos tareas. La primera de ellas tuvo como objetivo poner en práctica algunos de los conceptos relacionados con el reconocimiento de formas, mientras que en la segunda se realizó un estudio de los modelos

de matrices de covarianzas para la estimación de componentes gaussianas a través del algoritmo EM.

En relación a la segunda parte, la primera tarea consistió en realizar el diseño de una red bayesiana a partir de un problema, así como llevar a cabo su verificación. La segunda tarea puso en práctica el empleo de la técnica de d-separación, mientras que la tercera y última tarea se centró en la aplicación de procesos de inferencia sobre redes bayesianas.

Por último, a la tercera parte de la asignatura también se le asignaron tres tareas: una de aprendizaje paramétrico, otra de aprendizaje de redes bayesianas basados en tests de independencia condicional y otra de aprendizaje mediante técnicas *score+search*.

Relación de la asignatura con la Tesis Doctoral

Es innumerable la cantidad de trabajos relacionados con HEVC que hacen uso de clasificadores y técnicas de sistemas inteligentes. Uno de los usos más extendidos consiste en acelerar el particionamiento en bloques de la imagen mediante el uso de estas herramientas. Es por este y por otros motivos que resulta interesante conocer las técnicas y conocimientos provistos en esta asignatura. En este sentido, podría considerarse utilizar estos conceptos en el algoritmo de pre-análisis propuesto a lo largo de la Tesis Doctoral.

2.5. Modelado y Evaluación de Sistemas

La asignatura Modelado y Evaluación de Sistemas es impartida por los profesores Dr. D. Rafael Casado González, Dr. D. Aurelio Bermúdez Marín y Dr. D. Luis Orozco Barbosa del grupo de *Redes y Arquitecturas de Altas Prestaciones* (RAAP). Su principal objetivo es presentar los métodos y técnicas que permiten realizar la evaluación de un sistema, siendo la simulación la más recurrida de ellos.

Tras una introducción de los principales conceptos relacionados con el modelado y la evaluación de sistemas, esta asignatura presenta las métricas más empleadas en este sentido. Posteriormente, se introduce uno de los modelos matemáticos más empleados en la evaluación de sistemas, las cadenas de Markov, las cuales dan paso a la explicación de la teoría de colas.

En un segundo lugar, se pone en práctica estos y otros conceptos mediante el uso de los simuladores NS-2 y Riverbed Modeler. Con ellos, se diseña y modela alguna red de interconexión sencilla y diversas redes cableadas de comunicaciones.

Por último, se muestran las técnicas de modelado, especificación y control de sistemas dinámicos. Estas técnicas vienen acompañadas del uso de la herramienta Matlab Simulink, que permiten simular los modelos realizados y analizar los resultados obtenidos en función del tiempo.

Trabajo realizado en la asignatura

La asignatura tiene un marcado carácter práctico. Por este motivo, contempla la realización de múltiples tareas y prácticas a lo largo de la misma.

En relación a la primera parte, se realizó una serie de ejercicios teórico-prácticos relacionados con la teoría de la probabilidad, la teoría de colas y las redes de interconexión. Posteriormente, y mediante el simulador QNAP, se realizó el modelo de una cola simple de tipo M/M/1 y se analizaron sus prestaciones en función de diversos parámetros.

La segunda parte de la asignatura contempla la realización de diversas prácticas con los simuladores NS-2 y Riverbed Modeler previamente mencionados. A través de una serie de simulaciones, se evaluaron nuevamente las prestaciones de una cola M/M/1 y, además, las de diversas redes de comunicación cableadas.

Por último, a raíz de una introducción a la herramienta Matlab Simulink, se modeló una serie de problemas y sistemas de carácter dinámico, analizando su comportamiento a lo largo del tiempo.

Relación de la asignatura con la Tesis Doctoral

Los conocimientos adquiridos en esta asignatura son imprescindibles para todos aquellos campos de la investigación en los que es necesario evaluar las prestaciones de un determinado sistema o compararlas con otro dado. En el caso de la Tesis Doctoral propuesta, es requisito evaluar el rendimiento del codificador de HEVC no sólo en términos de eficiencia de codificación, sino también en el de tiempo de ejecución. Es por ello que resulta de gran importancia conocer la metodología y las técnicas necesarias para realizar una correcta evaluación de dicho sistema.

2.6. Modelos para el Análisis y Diseño de Sistemas Concurrentes

La asignatura se encuentra impartida por Dr. D. Valentín Valero Ruíz, Dr. D. Fernando Cuartero Gómez y Dr. D. Gregorio Díaz Descalzo del grupo de *Sistemas*

Concurrentes y Tiempo Real (RETICS). Tal y como se deja entrever por su nombre, el principal objetivo de la asignatura es presentar algunas de las técnicas de modelado de sistemas concurrentes existentes en el estado del arte, así como analizar algunas de sus propiedades más importantes.

En la primera parte, se hace una profunda descripción de las redes de Petri y sus características. A partir de ello, se explican las distintas propiedades de vivacidad y seguridad que puede cumplir un sistema concurrente y que pueden extraerse a partir de su modelo.

La segunda parte, en cambio, se centra en la especificación de sistemas concurrentes mediante álgebras de procesos. Así, se estudian lenguajes como *Communicating Sequential Processes* (CSP) y *Calculus of Communicating Systems* (CCS) para la especificación de los modelos, y la lógica de Hennessy-Milner para comprobar si un determinado sistema cumple una serie de propiedades.

Por último, y a partir de los conceptos anteriores, se introduce los autómatas temporizados para añadir relojes y propiedades temporales a los modelos.

Trabajo realizado en la asignatura

La aplicación de los conocimientos adquiridos en la asignatura se realiza de manera práctica, a la vez que se emplean conceptos teóricos para comprobar diversas propiedades de los sistemas. Así, la primera parte supuso modelar cuatro casos prácticos empleando redes de Petri mediante la herramienta Tina Toolbox. En la segunda parte se propuso la elaboración de un caso práctico empleando álgebras de procesos, apoyándose en la herramienta Concurrency Workbench, con la cual han de comprobarse ciertas propiedades mediante la lógica de Hennessy-Milner. Por último, también se modeló la solución a un problema de carácter real con autómatas temporales a través de la herramienta UPPAAL.

Relación de la asignatura con la Tesis Doctoral

El modelado formal de sistemas mediante los lenguajes y técnicas adquiridos en esta asignatura permiten extraer propiedades y características de los mismos que de otro modo no sería posible. Además, son la única alternativa posible a la hora de realizar la validación completa de un sistema. Por ello, los modelos formales permiten establecer una base sólida sobre la cual demostrar la validez de cualquier algoritmo implementado durante el desarrollo de la Tesis Doctoral.

2.7. Grid Computing

Esta asignatura es impartida en lengua inglesa por los profesores Dr. D. Fernando López Pelayo y Dra. D.^a María Emilia Cambroner Piqueras del grupo de *Sistemas Concurrentes y Tiempo Real* (RETICS). En línea de la anterior, esta asignatura se centra en el uso de métodos formales, salvo que aplicados a una serie de campos específicos.

Por un lado, en la primera parte se detalla el uso de especificaciones formales y de diagramas para definir y verificar servicios web bajo restricciones temporales. De manera similar, también se introduce una especificación formal enfocada a la descripción y verificación de contratos electrónicos.

Por otra parte, se da paso al lenguaje formal *Reasoning On Stochastic Algebras* (ROSA), el cual comparte características con álgebras de procesos como CCS. Este lenguaje es aplicado en múltiples ámbitos, como pueden ser un codificador MPEG-2 o grandes sistemas de computación, permitiendo hacer mediciones de rendimiento y analizar las vías de ejecución de los algoritmos correspondientes. Con ello, es posible detectar cuellos de botella, o puntos de posible paralelización y aceleración de los algoritmos.

Trabajo realizado en la asignatura

La asignatura consta de dos trabajos. En primer lugar, se hubo de realizar la especificación de un contrato cualquiera mediante las técnicas aprendidas, así como la validación de sus principales cláusulas y propiedades temporales. En segundo lugar, hubo de realizarse una especificación formal empleando ROSA de algún aspecto relacionado con la Tesis Doctoral. En este caso, se llevó a cabo la especificación del codificador de referencia de HEVC, del cual se hizo una descripción general de los módulos que lo componen y de su algoritmo de particionamiento en bloques.

Relación de la asignatura con la Tesis Doctoral

El lenguaje de especificación introducido en esta asignatura, ROSA, permite extraer características muy interesantes de los algoritmos paralelos. Entre ellas, la más importante es poder conocer el camino crítico de un algoritmo dado, lo que permite poder analizar aquellas partes del mismo que pueden ser optimizadas para reducir el tiempo total de ejecución. Por ello, el modelo llevado a cabo como trabajo de la asignatura permitió, además de tener un conocimiento mucho más profundo del estándar, examinar posibles puntos del codificador que podrían resultar de mayor complejidad y que podían convertirse en foco de atención del algoritmo desarrollado en la Tesis Doctoral.

Capítulo 3

FUNDAMENTOS DEL ESTÁNDAR HEVC

En los últimos treinta años, ha habido un marcado crecimiento del tráfico y del consumo de contenido multimedia, especialmente debido al desarrollo de las redes de comunicaciones y a la reciente aparición de los dispositivos móviles. Con el objetivo de reducir la cantidad de información que es necesario almacenar y transmitir, en este período de tiempo se han elaborado diversos estándares de compresión de vídeo. Generalmente, estos procesos de estandarización definen la sintaxis que debe tener la información codificada y cómo decodificarla con el objetivo de reconstruir los datos originales. Así, un formato de vídeo estandarizado tiene una serie de ventajas con respecto a aquéllos que no lo son o que son formatos propietarios [13]. Entre ellas, la más importante sin duda es la interoperabilidad entre codificadores y decodificadores de distintos fabricantes.

En esta línea, existen dos organizaciones que se dedican exclusivamente a desarrollar estándares de vídeo, el *International Telecommunication Union-Telecommunication (ITU-T) Video Coding Experts Group (VCEG)* y el *International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Moving Picture Experts Group (MPEG)*. Si bien en el pasado estas organizaciones se dedicaron a desarrollar sus estándares de manera separada, actualmente han creado acuerdos de trabajo mutuo en el contexto de *Joint Collaborative Team on Video Coding (JCT-VC)* con expertos de ambos grupos. La Figura 3.1 muestra el trabajo de éstos a lo largo del tiempo.

De esta manera, el estándar HEVC [6] fue llevado a cabo por el JCT-VC, cerrando la primera versión en enero de 2013. Con el objetivo de dar soporte a escenarios y extensiones de rango adicionales, se especificó una nueva versión en abril de 2015, la cual se corresponde con la última disponible en el momento de escritura de la presente memoria. La aparición de este estándar vino motivada por la necesidad de mejorar la eficiencia de compresión de su antecesor, el estándar H.264/AVC, especialmente

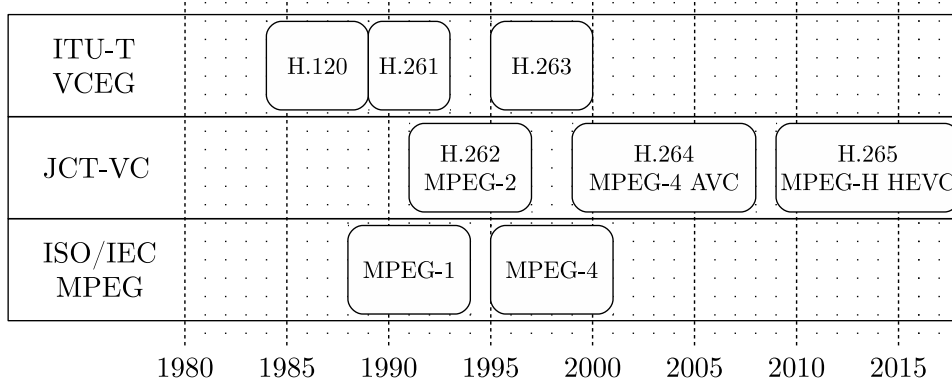


Figura 3.1: Línea temporal de los estándares de vídeo

acentuada con la llegada de resoluciones en torno al 4k y 8k, los modos multivista y el excesivo tráfico de vídeo en la red. Además, su diseño también fue enfocado al uso de arquitecturas de procesamiento paralelo.

La especificación del estándar no es en absoluto cerrada. El estándar describe la estructura del flujo de datos y su sintaxis, así como ciertas restricciones, de tal manera que ofrece total libertad para desarrollar las implementaciones específicas del codificador que lo utilice. El grupo JCT-VC, además, provee un software de referencia que es tomado como base en proyectos de carácter científico, así como un conjunto de datos de prueba para realizar comparaciones entre éste y diversas propuestas.

3.1. Consideraciones de diseño

Como se introdujo en el apartado anterior, el estándar HEVC está diseñado para mejorar notablemente la eficiencia de codificación de estándares anteriores y para ofrecer, entre otras cosas, mayor facilidad de integración, resistencia ante pérdidas de datos y la posibilidad de ser implementado en arquitecturas paralelas [14]. A pesar de todas estas mejoras, la Figura 3.2 muestra el diagrama de bloques típico de un *encoder* válido basado en HEVC, en el que se aprecia claramente cómo se sigue utilizando una aproximación híbrida (predicción inter/intra y transformación 2D) similar a la de otros estándares existentes desde H.261. Esto se debe a que las modificaciones más relevantes se han realizado dentro de cada uno de los módulos que lo componen, y no tanto a nivel estructural.

Como se observa en la figura, cada imagen que forma una secuencia de vídeo es dividida en regiones de bloque de tamaño idéntico. La primera imagen de la secuencia (y cada primera imagen después de un punto de acceso aleatorio) se codifica utilizando únicamente predicción intra, es decir, sin dependencias entre imágenes. Para el resto de las imágenes se utiliza predicción temporal o inter, la cual consiste básicamente en escoger el vector de movimiento de cada bloque que minimiza el residuo generado con respecto a la región de otra imagen denominada *frame* de referencia.

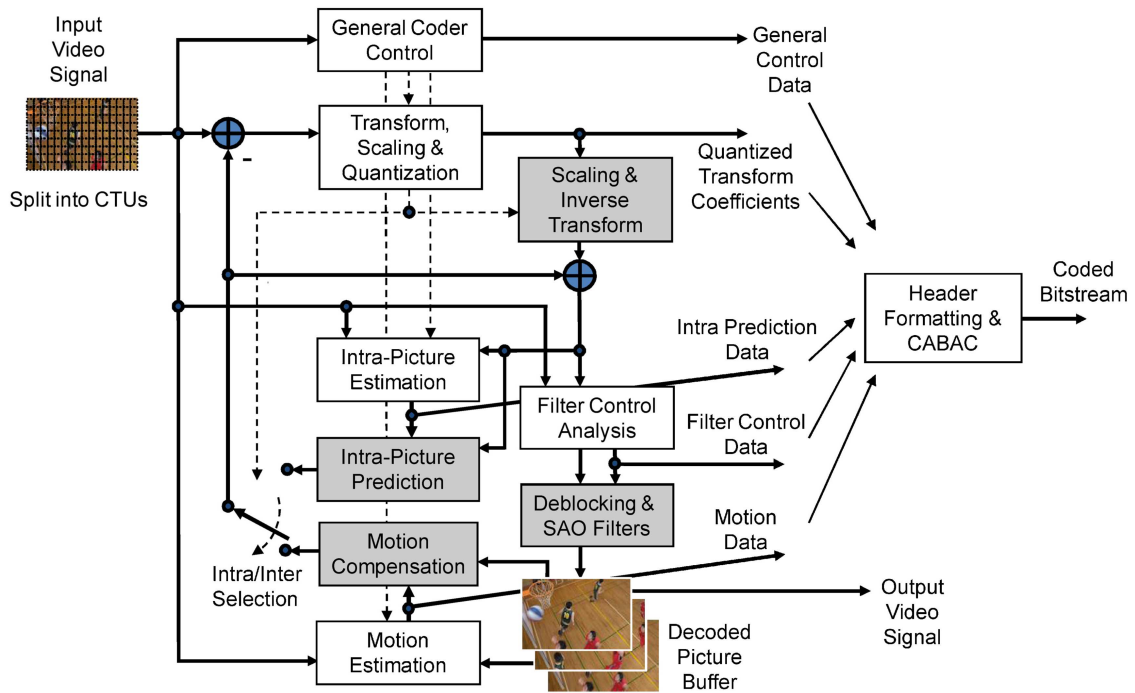


Figura 3.2: Diagrama de bloques de un codificador basado en HEVC [14]

Nótese que las partes sombreadas en gris se corresponden con módulos constituyentes del decodificador. Esto se debe a que el codificador también los ha de incorporar para reconstruir las señales que codifica. De esta manera, las referencias temporales se realizan sobre los *frames* reconstruidos en vez de los originales, anulando así cualquier desviación entre *encoder* y *decoder*. Estos *frames* de referencia son la salida del módulo de los *in-loop filters* a los que se le alimenta la suma del residuo codificado y la predicción para suavizar los posibles artefactos generados en la codificación en bloques.

3.2. División de un frame en unidades básicas

Al igual que ocurre en estándares anteriores como H.264/AVC, donde un *frame* se divide en bloques cuadrados de tamaño fijo, en HEVC una imagen se divide en bloques denominados *Coding Tree Units* (CTU). El estándar soporta un tamaño variable de CTU entre 16×16 , 32×32 y 64×64 píxeles, pudiendo ser parametrizado en el momento de la codificación en función de las limitaciones computacionales y de memoria existentes. Este aumento en el tamaño de la unidad básica de división frente a otros estándares tiene como objetivo mejorar la codificación de contenidos en alta resolución.

Adicionalmente, HEVC define una estructura en forma de *quadtrees* para subdividir los CTUs, de tal manera que el espacio definido por los mismos puede dividirse a su vez en cuatro subespacios de igual tamaño [15]. A estas subdivisiones se les denomina *Coding Units* (CUs). Este proceso de división es iterativo y puede repetirse hasta que un CU alcance el tamaño mínimo definido por la configuración o, en cualquier caso, un

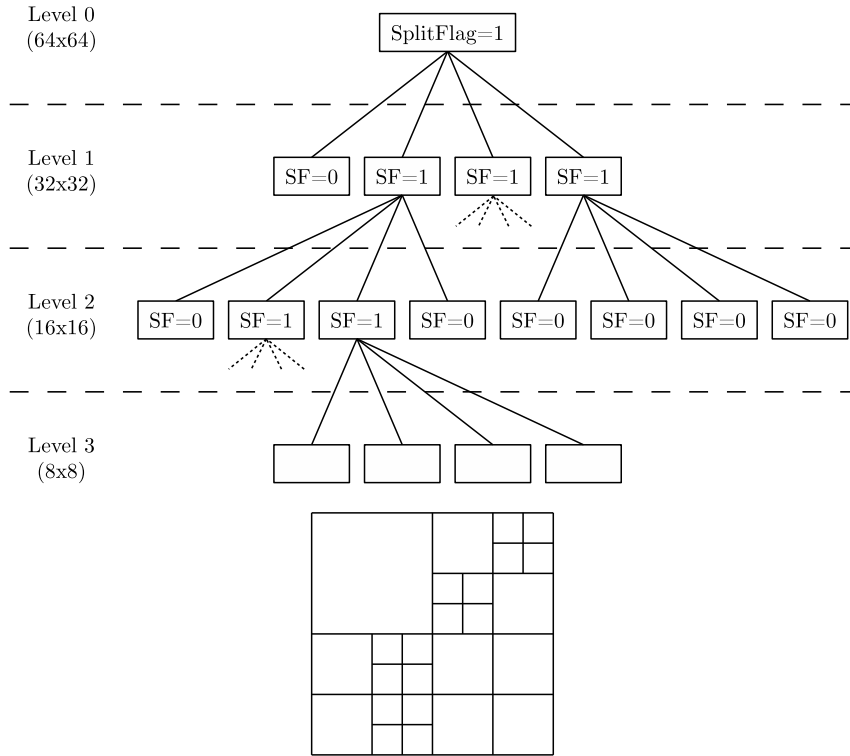


Figura 3.3: División de un CTU en CUs siguiendo una estructura de *quadtree*

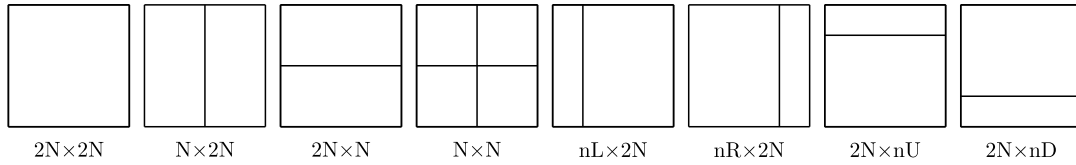


Figura 3.4: Esquemas de particionamiento de un CU en PUs

tamaño de 8×8 . Un ejemplo de la partición de un CTU en subdivisiones menores queda reflejado en la Figura 3.3, donde el *SplitFlag* es un elemento de sintaxis que indica si un CTU/CU se subdivide o no.

Bajo esta estructura, cada CU puede dividirse a su vez en *Prediction Units* (PU) y en una estructura de árbol de *Transform Units* (TU), los cuales son detallados a continuación.

Prediction Units (PU)

Los PUs son los elementos de sintaxis que contienen información relativa a la predicción, correspondiendo la decisión de codificarlo empleando predicción intra (espacial) o inter (temporal) al CU que lo contiene. El área a predecir puede dividirse en alguna de las posibles particiones representadas en la Figura 3.4.

En caso de realizarse predicción intra, el tamaño del PU es el mismo que el del CU que lo contiene para cualquier tamaño de CU, lo que se corresponde con una

partición del tipo $2N \times 2N$. Como única excepción, cuando el tamaño del CU es el mínimo permitido, un flag de codificación permite escoger entre dicho particionamiento o subdividir la región en cuatro PUs empleando el esquema $N \times N$. En este último caso, los PUs pueden alcanzar un tamaño de 4×4 píxeles.

Por otro lado, cuando se produce una predicción de tipo inter, el área del CU en cuestión puede ser particionada en uno, dos o cuatro PUs, siendo este último esquema posible únicamente cuando el tamaño de CU se corresponde con el mínimo permitido y los PUs correspondientes son de tamaño superior a 4×4 . Por tanto, a diferencia de la predicción tipo intra, los PUs no pueden alcanzar dicho tamaño. Para el caso de la división en dos PUs, cabe destacar la introducción en este estándar del llamado *Asymmetric Motion Partitioning* (AMP) que consiste en la partición de un CU en dos bloques de distinto tamaño, lo cual sólo es permitido cuando el tamaño de CU es superior o igual a 16×16 píxeles. Como restricción adicional, y con el objetivo de minimizar el coste en memoria, el estándar no permite los tamaños 8×4 y 4×8 para otro tipo que no sea la predicción unidireccional.

Para el tipo de predicción inter, a un PU le es asignado uno o dos vectores de movimiento y, a su vez, uno o dos índices de referencia, respectivamente.

Transform Units (TU)

El residuo generado por la predicción es codificado utilizando bloques sobre los que se les aplica una transformada. Estos bloques, que no tienen por qué coincidir con los PUs, se denominan *Transform Units* (TU) y tienen forma cuadrada. Su codificación sigue una estructura de árbol muy parecida a la ya representada en la Figura 3.3 cuya raíz se encuentra a nivel de CU, pudiendo alcanzar un tamaño mínimo de 4×4 . Esta estructura recibe la denominación de *Residual Quadtree* (RQT).

Para cada tamaño de TU 4×4 , 8×8 , 16×16 y 32×32 se define una función basada en la transformada de coseno discreta (*Discrete Cosine Transform*, DCT) y en el caso concreto del tamaño 4×4 se define una función alternativa basada en la transformada de seno discreta (*Discrete Sine Transform*, DST).

La Figura 3.5 muestra un ejemplo realizado sobre la secuencia *Traffic* de tamaño $4k \times 2k$. En dicho ejemplo, se muestra un fragmento de un *frame* que ha sido particionado en CUs (en blanco) y estos a su vez en TUs (en rojo).

3.3. División de un frame en unidades mayores

Como hemos visto en la Sección 3.2, existen unos bloques básicos de tamaño fijo llamados CTU en los que podemos dividir una imagen. Sin embargo, agrupar estos

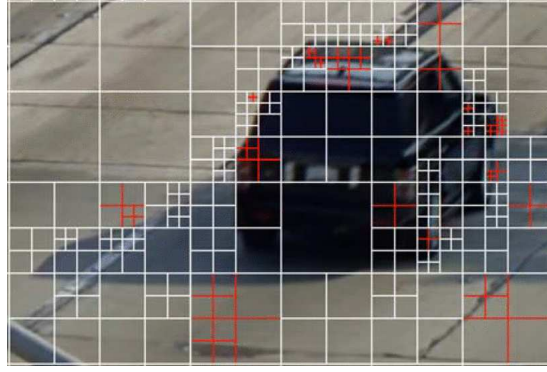


Figura 3.5: Fragmento en detalle de la partición de un *frame* en CUs y TUs

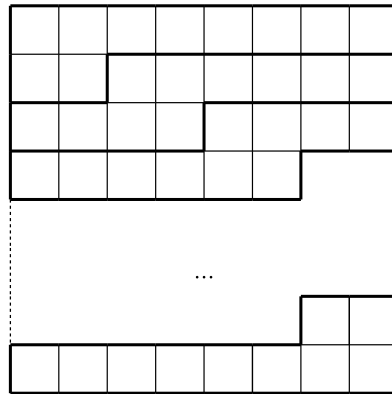


Figura 3.6: División de un *frame* en *slices*

bloques en una entidad tan grande como es el *frame* conlleva una serie de desventajas. La más importante a mencionar es la baja tolerancia a fallos o pérdidas de datos. Es por ello que es posible utilizar los llamados *slices*.

Un *slice* es una secuencia de CTUs que son procesados en orden de rasterización. Así, un *frame* puede estar dividido en uno o más *slices*. La Figura 3.6 muestra este tipo de división.

Los *slices* son autocontenidos en el sentido de que, suponiendo disponible la información de la secuencia y de la imagen a decodificar, su sintaxis puede ser interpretada desde el flujo de datos. En otras palabras, el área que ocupa puede ser correctamente decodificada, a excepción del *in-loop filtering* realizado sobre los bordes de los *slices*. Esto implica, sin embargo, no utilizar ningún tipo de información de otros *slices* en el mismo *frame*, incluyendo información para la predicción inter o intra.

Podemos distinguir tres tipos de *slice* según el tipo de información que puede codificar:

- *I slice*: se corresponde con un *slice* en el que todos los CUs son codificados utilizando únicamente predicción intra.

- *P slice*: además de los tipos de predicción admitidos por un *slice* de tipo I, los CUs de esta clase pueden ser codificados utilizando predicción inter con un máximo de un vector de movimiento.
- *B slice*: adicionalmente a las capacidades de un *slice* de tipo P, algunos CUs pertenecientes a este tipo de *slices* pueden ser codificados utilizando como mucho dos vectores de movimiento. En este caso, los *frames* de referencia se organizan en dos listas denominadas L0 y L1.

Como se ha mencionado anteriormente, el objetivo principal de los *slices* es hacer posible la resincronización tras una pérdida de datos. En este sentido, en muchas ocasiones se restringe el número máximo de bits por cada *slice*, limitando el número de CTUs en el mismo. Esto resulta especialmente adecuado para una transmisión en paquetes sobre una red de comunicaciones.

Por otra parte, el estándar HEVC ha introducido como novedad los llamados *tiles*, los cuales son regiones rectangulares autocontenidas e independientes las unas de las otras. Un *slice* puede contener múltiples *tiles* y viceversa. Su propósito principal es el de ser utilizado en técnicas de paralelización tanto en procesos de codificación como de decodificación, por lo que se desarrolla más en detalle en la Sección 3.8.

3.4. Predicción intra

La predicción intra o espacial es aquélla que se realiza empleando únicamente información del *frame* actual, es decir, no existen referencias a otros *frames* ya codificados, sino que la información se toma de los CUs vecinos.

El estándar que estamos tratando define 33 orientaciones distintas hacia las cuales puede realizarse una predicción de este tipo. Estas orientaciones han sido representadas en la Figura 3.7. De manera alternativa a estas direcciones, el estándar también define el modo *planar* que asume un degradado vertical u horizontal desde los bordes, y la predicción DC, la cual toma una superficie plana con el valor de la media de las muestras circundantes.

Para eliminar las discontinuidades entre las fronteras de los bloques en los modos DC, direccional totalmente horizontal y direccional totalmente vertical, las muestras limítrofes son suavizadas mediante el uso de un filtro. A su vez, para los modos direccionales que se acercan a la horizontal o a la vertical también se les aplica un filtro a nivel de bloque (dependiendo de su tamaño) con tal de evitar cambios abruptos en los valores.

Por otro lado, para un PU determinado, se determinan los tres modos más probables (*Most Probable Modes*, MPM) en función de los PUs vecinos. De esta manera, si el

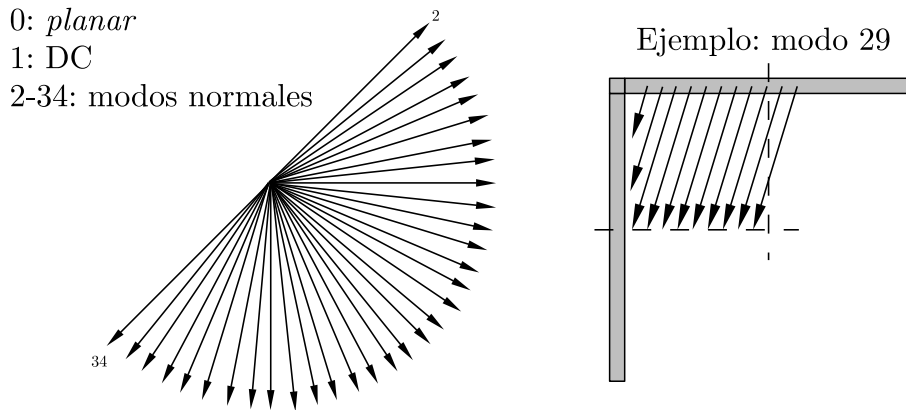


Figura 3.7: Modos y orientaciones para la predicción intra

mejor modo encontrado coincide con alguno de estos tres modos predichos, tan sólo será necesario señalar su índice correspondiente en el flujo de datos. En caso contrario, habrá que codificar el índice completo. Esta funcionalidad está diseñada para mejorar la tasa de bits.

3.5. Predicción inter

La predicción inter se corresponde con aquélla que busca la mejor coincidencia del PU actual en las áreas de búsqueda establecidas en el conjunto de *frames* de referencia ya codificados y reconstruidos. Es por ello que también se le conoce como predicción temporal.

El resultado de este proceso es un vector de movimiento (*Motion Vector*, MV) que apunta a la zona del *frame* de referencia correspondiente donde la diferencia es mínima y la eficiencia conseguida es máxima. Estos vectores de movimiento no tienen por qué ser necesariamente enteros. De hecho, HEVC define una interpolación sub-píxel para soportar MVs con unidades de un cuarto de la distancia entre píxeles. Este filtro de interpolación se aplica en un único paso y depende del propio vector de movimiento:

- Para posiciones de un medio de precisión, se define un filtro de ocho muestras (*eight-tap filter*) cuyos coeficientes vienen reflejados en la Tabla 3.1 según la fila *hfilter*.
- Para posiciones de un cuarto de precisión, se define un filtro de siete muestras (*seven-tap filter*). Sus coeficientes están representados en la Tabla 3.1 conforme a la fila *qfilter*.

Estos valores contenidos en la tabla anteriormente mencionada han sido parcialmente derivados a partir de la transformada de coseno discreta.

Cuadro 3.1: Coeficientes aplicados en los filtros *eight-tap* y *seven-tap* de interpolación sub-píxel

Índice i	-3	-2	-1	0	1	2	3	4
hfilter[i]	-1	4	-11	40	40	-11	4	1
qfilter[i]	-1	4	-10	58	17	-5	1	

$A_{-1,-1}$				$A_{0,-1}$	$a_{0,-1}$	$b_{0,-1}$	$c_{0,-1}$	$A_{1,-1}$				$A_{2,-1}$
$A_{-1,0}$				$A_{0,0}$	$a_{0,0}$	$b_{0,0}$	$c_{0,0}$	$A_{1,0}$				$A_{2,0}$
$d_{-1,0}$				$d_{0,0}$	$e_{0,0}$	$f_{0,0}$	$g_{0,0}$	$d_{1,0}$				$d_{2,0}$
$h_{-1,0}$				$h_{0,0}$	$i_{0,0}$	$j_{0,0}$	$k_{0,0}$	$h_{1,0}$				$h_{2,0}$
$n_{-1,0}$				$n_{0,0}$	$p_{0,0}$	$q_{0,0}$	$r_{0,0}$	$n_{1,0}$				$n_{2,0}$
$A_{-1,1}$				$A_{0,1}$	$a_{0,1}$	$b_{0,1}$	$c_{0,1}$	$A_{1,1}$				$A_{2,1}$
$A_{-1,2}$				$A_{0,2}$	$a_{0,2}$	$b_{0,2}$	$c_{0,2}$	$A_{1,2}$				$A_{2,2}$

Figura 3.8: Posiciones enteras y sub-píxel en el filtro de interpolación

En la Figura 3.8, las posiciones etiquetadas con letras mayúsculas, $A_{i,j}$, representan muestras luma en posiciones enteras, mientras que las demás posiciones representan posiciones sub-píxel cuyos valores han de ser interpolados a partir de los primeros. Por ejemplo, las muestras $a_{0,0}$, $b_{0,0}$, $c_{0,0}$, $d_{0,0}$, $h_{0,0}$ y $n_{0,0}$ pueden obtenerse a partir de la Ecuación 3.1, donde B es la profundidad de bits de las muestras (típicamente 8):

$$\begin{aligned}
 a_{0,0} &= ((\sum_{i=-3}^3 (A_{i,0} \cdot qfilter[i])) \gg (B-8)) \gg 6 \\
 b_{0,0} &= ((\sum_{i=-3}^4 (A_{i,0} \cdot hfilter[i])) \gg (B-8)) \gg 6 \\
 c_{0,0} &= ((\sum_{i=-2}^4 (A_{i,0} \cdot qfilter[1-i])) \gg (B-8)) \gg 6 \\
 d_{0,0} &= \left(\left(\sum_{j=-3}^3 (A_{0,j} \cdot qfilter[j]) \right) \gg (B-8) \right) \gg 6 \\
 h_{0,0} &= \left(\left(\sum_{j=-3}^4 (A_{0,j} \cdot hfilter[j]) \right) \gg (B-8) \right) \gg 6 \\
 n_{0,0} &= \left(\left(\sum_{j=-2}^4 (A_{0,j} \cdot qfilter[1-j]) \right) \gg (B-8) \right) \gg 6
 \end{aligned} \tag{3.1}$$

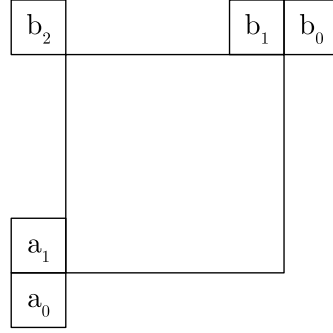


Figura 3.9: Posiciones de los candidatos espaciales del modo *Merge*

Por su parte, los valores etiquetados como $e_{0,0}$, $f_{0,0}$, $g_{0,0}$, $i_{0,0}$, $j_{0,0}$, $k_{0,0}$, $p_{0,0}$, $q_{0,0}$ y $r_{0,0}$ pueden derivarse a partir de aplicar el filtro verticalmente utilizando las muestras anteriormente calculadas, según la Ecuación 3.2:

$$\begin{aligned}
 e_{0,0} &= \left(\sum_{v=-3}^3 (a_{i,v} \cdot qfilter[v]) \right) >> 6 \\
 f_{0,0} &= \left(\sum_{v=-3}^3 (b_{i,v} \cdot qfilter[v]) \right) >> 6 \\
 g_{0,0} &= \left(\sum_{v=-3}^3 (c_{i,v} \cdot qfilter[v]) \right) >> 6 \\
 i_{0,0} &= \left(\sum_{v=-3}^4 (a_{i,v} \cdot qfilter[v]) \right) >> 6 \\
 j_{0,0} &= \left(\sum_{v=-3}^4 (b_{i,v} \cdot qfilter[v]) \right) >> 6 \\
 l_{0,0} &= \left(\sum_{v=-3}^4 (c_{i,v} \cdot qfilter[v]) \right) >> 6 \\
 p_{0,0} &= \left(\sum_{v=-2}^4 (a_{i,v} \cdot qfilter[1-v]) \right) >> 6 \\
 q_{0,0} &= \left(\sum_{v=-2}^4 (b_{i,v} \cdot qfilter[1-v]) \right) >> 6 \\
 r_{0,0} &= \left(\sum_{v=-2}^4 (c_{i,v} \cdot qfilter[1-v]) \right) >> 6
 \end{aligned} \tag{3.2}$$

Merge Mode

El *encoder* puede decidir no realizar la estimación de movimiento. En tal caso, HEVC incluye el llamado *Merge Mode* para derivar la información de movimiento a partir de bloques vecinos. De esta manera, se transmitiría el índice del candidato y los índices de la lista y del *frame* de referencia del mismo, evitando tener que enviar toda los datos relativos a un vector de movimiento.

El conjunto de posibles candidatos en el modo *Merge* está compuesto de vecinos espaciales, un candidato temporal y un candidato generado. La Figura 3.9 muestra la posición de los cinco candidatos espaciales, cuya disponibilidad se comprueba en el orden a_1 , b_1 , b_0 , a_0 , b_2 . Dicha disponibilidad dependerá de si el candidato en esa posición fue codificado como *inter* y de si se encuentra dentro de los límites del *frame/slice*. Para el candidato temporal, se toma el PU colocado en la posición inferior derecha del *frame* de referencia, si estuviera disponible y, si no, se toma el de la posición central.

Cabe mencionar también que el conocido modo *Skip* en otros estándares anteriores se corresponde con un caso especial del modo *Merge* en el cual todos los flags de codificación equivalen a cero.

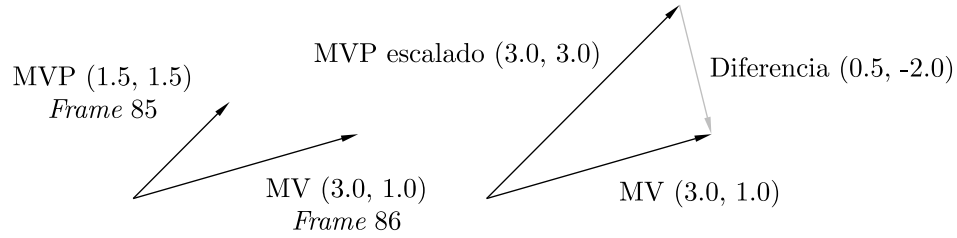


Figura 3.10: Escalado del predictor de movimiento de un PU vecino en otro *frame*

Predictores de movimiento

Si el *encoder* ha decidido no utilizar el modo *Merge*, éste habrá calculado un MV. Este vector es codificado calculando su diferencia con el denominado predictor de movimiento (*Motion Vector Predictor*, MVP), reduciendo así (en general) la cantidad de bits a transmitir al decodificador.

El conjunto de posibles predictores se forma a partir de los vectores pertenecientes a los bloques del conjunto que utiliza el modo *Merge* según la Figura 3.9. Sin embargo, se escoge un candidato espacial del conjunto a_0 , a_1 y otro del conjunto b_0 , b_1 , b_2 , en ese orden. En caso de que al menos uno de ellos no estuviera disponible, se incluiría un candidato temporal o, en caso de tampoco estar disponible, el vector nulo. El proceso por el cual el *encoder* selecciona finalmente uno de ellos se denomina *Advanced Motion Vector Prediction* (AMVP).

La razón del reducido número de candidatos es, en primer lugar, que el codificador puede codificar el vector de movimiento como su diferencia con el predictor, por lo que no es necesario disponer de un gran número de éstos. Además, se necesita realizar la estimación de movimiento, la cual es una de las operaciones más costosas dentro del codificador, por lo que un incremento en el número de candidatos influirá directamente en su complejidad computacional.

En caso de que el índice de referencia del PU vecino no equivalga al del PU actual, el vector de movimiento se escala en una magnitud igual a la diferencia temporal entre el *frame* actual y el de referencia. Este hecho viene reflejado en el ejemplo de la Figura 3.10.

3.6. Transformada, cuantización y entropía

HEVC transforma el residuo generado por la predicción de una manera muy similar a como lo hacían estándares anteriores. El bloque que contiene el residuo es dividido en múltiples particiones cuadradas según fue descrito en la Sección 3.2, cuyos posibles tamaños son 4×4 , 8×8 , 16×16 y 32×32 .

Las transformadas bidimensionales son calculadas mediante el cálculo de transformadas unidimensionales en la dirección horizontal y vertical. Los elementos de las matrices utilizadas en la transformación han sido obtenidas a partir de las funciones de la DCT, definiendo una para cada tamaño de bloque. Para el caso de los bloques de tamaño 4×4 , el estándar permite utilizar la transformada DST. La justificación de esta característica parte del hecho de que la amplitud del residuo tiende a incrementar conforme la distancia a los límites de las muestras usadas en la predicción se hace mayor, lo cual es reflejado de mejor manera por este tipo de transformada. Además, su coste computacional es de la magnitud de la DCT, proporcionando un 1 % de reducción en *bit-rate* [14].

En cuanto a cuantización se refiere, HEVC utiliza el mismo esquema controlado por el parámetro de cuantización (*Quantization Parameter*, QP) que en estándares anteriores.

En relación a la codificación de entropía, el estándar tan sólo define el método *Context-Adaptive Binary Arithmetic Coding* (CABAC), en vez de definir dos distintos como ocurría en H.264/AVC.

3.7. Filtros in-loop

En HEVC se aplican dos filtros de postprocesado sobre las muestras reconstruidas antes de almacenarlas en el buffer de imágenes decodificadas (*Decoded Picture Buffer*, DPB). Estos filtros se denominan *Deblocking Filter* (DBF) y *Sample Adaptive Offset* (SAO), respectivamente.

El DBF pretende reducir los artefactos en forma de bloque generados por la manera en que se realiza la codificación en este estándar. Por ello, este filtro se aplica a las muestras limítrofes a los bordes de los bloques. El filtro SAO, en cambio, ha sido introducido en HEVC y se aplica adaptativamente a todas las muestras que cumplen ciertas condiciones.

Deblocking Filter (DBF)

El filtro DBF es aplicado a todas las muestras adyacentes al borde de un PU o de un TU, exceptuando el caso de que éste coincida con el borde de la imagen. Cabe hacer notar que se considera tanto los bordes de los PUs como los de los TUs debido a que estas estructuras no siempre están alineadas. Sin embargo, sólo se aplica a los bordes que se encuentran alineados a una rejilla de 8×8 para reducir la complejidad computacional del algoritmo.

Se definen tres intensidades a la hora de aplicar el filtro. Dados dos bloques P y Q , una intensidad 0 indica que éste no se aplica, una intensidad 2 se aplica cuando uno

de los bloques ha sido codificado como intra y la intensidad 1 se asigna cuando ocurre uno de los siguientes supuestos:

1. P o Q contienen, al menos, un coeficiente de la transformada no nulo.
2. Los índices de referencia de P y Q no son iguales.
3. Los vectores de movimiento de P y Q no son iguales.
4. La diferencia entre un componente del MV de P y de Q es mayor o igual a un píxel.

A partir de este nivel de intensidad se puede conocer una serie de umbrales y valores a utilizar en el filtrado, el cual se realiza primero horizontalmente para solventar los bordes verticales y, posteriormente, en su forma inversa.

Sample Adaptive Offset (SAO)

Éste es un proceso que modifica las muestras decodificadas añadiendo (de manera condicional) un valor a cada una de ellas. Su objetivo principal es aumentar la calidad de la imagen reduciendo los posibles artefactos derivados del proceso de codificación. El estándar define dos modos:

- *Band offset mode*. Depende de la amplitud de la muestra. En este modo, el rango de amplitudes es uniformemente dividido en 32 segmentos llamados bandas. Las bandas son agrupadas en conjuntos de 4 y se les añade un valor positivo o negativo, suavizando así áreas en las cuales estas bandas puedan generar artefactos.
- *Edge offset mode*. En este otro modo, se compara el valor de una muestra con dos de sus ocho vecinos usando uno de los cuatro patrones de degradado: horizontal, vertical y las dos diagonales. Según esta comparación, una muestra puede considerarse como zona monótona, mínimo local, máximo local o dos tipos de bordes, es decir, cinco tipos en total. Para cualquiera de los cuatro últimos, se le añade un valor a la muestra para corregirla.

3.8. Técnicas de paralelización

Con el objetivo de superar las limitaciones existentes en estándares anteriores relativas a la utilización de estrategias de paralelización, HEVC incluye dos herramientas enfocadas a facilitar el procesamiento en paralelo a alto nivel: *Wavefront Parallel Processing* (WPP) y *Tiles*. Ambas estrategias permiten la subdivisión de los *frames* en múltiples

0	1	2	3	12	13	14	21	22	23	24
4	5	6	7	15	16	17	25	26	27	28
8	9	10	11	18	19	20	29	30	31	32
33	34	35	36	41	42	43	47	48	49	50
37	38	39	40	44	45	46	51	52	53	54
55	56	57	58	63	64	65	69	70	71	72
59	60	61	62	66	67	68	73	74	75	76

Figura 3.11: Particionamiento de un *frame* en *tiles* y su correspondiente orden de procesamiento

particiones que pueden ser procesadas en paralelo. Estas particiones contienen CTUs que pueden guardar dependencias entre sí.

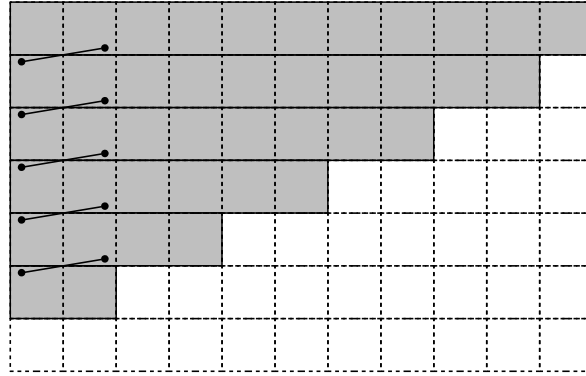
Cabe añadir que por motivos de diseño, no es posible utilizar ambas estrategias al mismo tiempo.

3.8.1. *Tiles*

Mediante la utilización de esta estrategia de paralelización, la imagen es dividida en grupos rectangulares de CTUs separados por unas fronteras verticales y horizontales de manera similar a como muestra la Figura 3.11. El número de *tiles* y la posición de sus bordes puede ser definido para la secuencia entera o por cada una de las imágenes que la componen. La única restricción existente viene dada por el hecho de que todos los CTUs en un *slice* deben pertenecer al mismo *tile*, o viceversa.

Estos bordes, de manera similar a como ocurre con los correspondientes de los *slices*, rompen las dependencias existentes, provocando así que los *tiles* puedan ser procesados de manera concurrente. Sin embargo, los filtros *in-loop* todavía pueden operar a través de distintos *tiles*. Esta estrategia de paralelización cambia, además, el orden en el que son procesados los CTUs.

Comparado con la opción de dividir una imagen en *slices*, los *tiles* tienen una mejor eficiencia de codificación debido a la forma de las particiones, es decir, los CTUs dentro de un *tile* poseen una mayor correlación. Además, el uso de *tiles* reduce la carga de cabeceras en comparación. Sin embargo, la tasa de pérdidas en calidad (ruido-distorsión) también se ve incrementada conforme el número de *tiles* utilizados aumenta, puesto que las dependencias se rompen y las probabilidades de entropía CABAC se reinician al principio de cada división.

**Figura 3.12:** Procesamiento en filas de WPP**Cuadro 3.2:** Comparación de las estrategias de paralelización disponibles en HEVC

Propiedad	<i>Slices</i>	<i>Tiles</i>	WPP
Pérdidas en codificación	Muy alta	Alta	Baja
Artefactos visuales en los bordes	Sí	Sí	No
Problemas en single-core	No	Sí	No
Escalabilidad	Media	Media	Media/Alta
Contempla regiones de interés	No	Sí	No

3.8.2. Wavefront Parallel Processing (WPP)

La estrategia de paralelización WPP toma cada fila de CTUs como una partición distinta. En contraposición con los *slices* y los *tiles*, no se rompe ninguna dependencia en los límites de dichas filas. De hecho, las probabilidades CABAC son propagadas desde (como mínimo) el segundo CTU de la fila anterior para reducir las pérdidas en la codificación en paralelo (Figura 3.12). Como ventaja adicional, un flujo de bits codificado con WPP puede ser convertido sin pérdidas a un flujo no paralelo, y viceversa.

Al activar WPP, un número de procesadores igual al de filas de CTUs puede trabajar en paralelo para procesarlas. Las restricciones anteriormente mencionadas, sin embargo, no permiten que cada fila comience a ser procesada desde el inicio de cada imagen. Por tanto, las filas tampoco terminarán de codificarse o decodificarse a la vez. Esto introduce un fenómeno denominado como *ramping inefficiencies* que se hace más notable cuando se utiliza un mayor número de procesadores.

3.8.3. Aplicaciones de cada estrategia de paralelización

De las subsecciones anteriores podemos extraer claramente que tanto *Tiles* como WPP tienen sus ventajas e inconvenientes. La Tabla 3.2 muestra un resumen de las principales propiedades de cada una de estas aproximaciones.

WPP es generalmente más adecuado para paralelizar tanto el *encoder* como el *decoder* por su mayor número de particiones por imagen a costa de una baja tasa de

pérdidas de compresión. Además, no genera artefactos visuales en las fronteras de las particiones. WPP también puede ser usado en aplicaciones *low-delay* donde es necesario que el codificador envíe las particiones ya codificadas tan pronto las tenga disponibles.

Por su parte, *Tiles* también puede ser utilizado para paralelizar tanto el codificador como el decodificador. Sin embargo, la cantidad de paralelismo disponible no es fija como ocurre en WPP, permitiendo al *encoder* ajustar el número de *tiles* a utilizar en función de los recursos computacionales disponibles. Puesto que, además, permite dividir la imagen en regiones rectangulares, esta estrategia de paralelización es más adecuada para codificar las llamadas regiones de interés (*Region Of Interest*, ROI).

En términos de rendimiento, *Tiles* ofrece un mayor grado de paralelización frente a WPP, si bien hace que la eficiencia de codificación se vea penalizada en mayor medida [16].

3.9. Perfiles, niveles y tiers

Con el objetivo de facilitar la interoperabilidad entre distintas aplicaciones y dispositivos, el estándar HEVC especifica una serie de perfiles (*profiles*), *tiers* y niveles (*levels*).

Un perfil define una serie de herramientas de codificación y algoritmos que pueden ser utilizados para generar un determinado flujo de bits, mientras que un nivel restringe ciertos parámetros relativos a dicho flujo, como el tamaño de la imagen o el *ratio* de compresión mínimo. Los *tiers*, por su parte, sólo fueron diseñados para diferenciar entre aplicaciones que necesitan de un mayor flujo de bits (*High tier*) de lo normal (*Main tier*). Así, un *decoder* que esté diseñado conforme a un determinado nivel y *tier* deberá poder decodificar cualquier flujo de bits válido de un menor nivel o *tier*.

La primera versión del estándar define tres perfiles enfocados a tres fines distintos:

- *Main profile*. Acepta el muestreo 4:2:0 con una profundidad de 8 bits por muestra. Está destinado a la codificación de vídeo general.
- *Main 10 profile (high efficiency)*. Igual al perfil *Main* pero tomando 10 bits por muestra. Adecuado para vídeos en que es necesario una mayor calidad visual.
- *Main Still Picture*. Similar al perfil *Main*, pero para codificar imágenes estáticas.

Las siguientes versiones del estándar definieron, a su vez, un mayor número de perfiles destinados a dar soporte a extensiones de rango, lo cual incluye muestreos como 4:4:4 o incluso una profundidad de 16 bits.

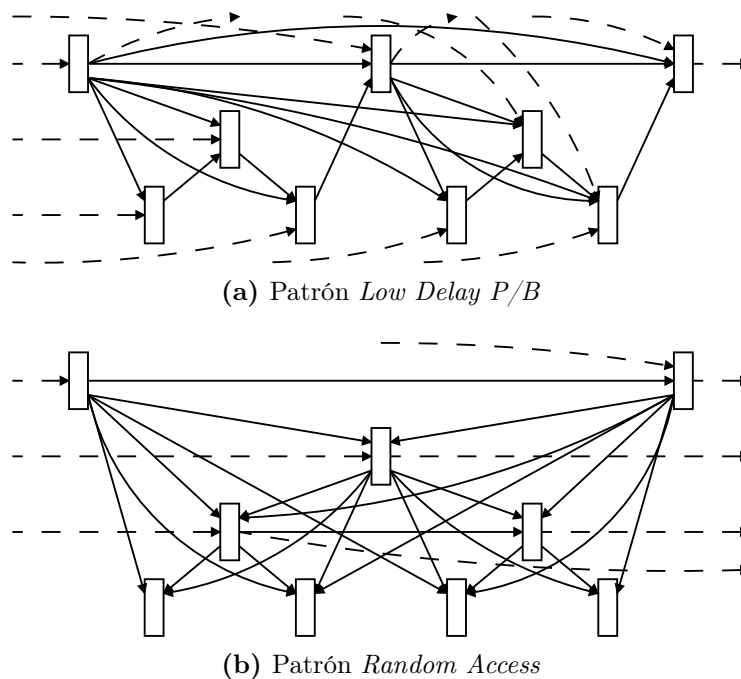


Figura 3.13: Patrones de codificación *Low Delay* y *Random Access*, y sus referencias

3.10. Patrones de codificación

Si bien el estándar no indica cómo deben organizarse los *frames* en una secuencia de vídeo o en qué orden han de codificarse, sí que podemos encontrar diversas nomenclaturas en la literatura e incluso en el propio software de referencia que provee JCT-VC que indican el patrón que siguen los cuadros y las referencias en una secuencia. Estos modos de codificación son los siguientes:

- *All Intra*. Todos los *frames* son codificados mediante *slices* tipo I. En otras palabras, no existen referencias hacia otros cuadros y, por tanto, no hay estimación de movimiento.
- *Low Delay P/B*. Los *frames* son codificados con *slices* tipo P o B, excepto el primero que es de tipo I. Por tanto, la secuencia tendrá la forma IPPP... o IBBB..., siendo las referencias únicamente hacia cuadros anteriores. Un ejemplo de este patrón puede apreciarse en la Figura 3.13a, donde las líneas discontinuas representan referencias a *frames* no presentes en el esquema representado.
- *Random Access*. En este caso, los *frames* pueden ser codificados y recibidos en el decodificador en orden distinto al de reproducción. Por tanto, las referencias pueden ser hacia *frames* pasados o futuros. Este modo utiliza una estructura jerárquica de *slices* tipo B de la manera en que se representa en la Figura 3.13b.

Capítulo 4

TRABAJO DE INVESTIGACIÓN: ALGORITMO DE PRE-ANÁLISIS PARA HEVC

Tal y como se ha puesto de manifiesto en secciones anteriores, el desarrollo del estándar de vídeo HEVC [6] vino motivado por la creciente demanda de contenidos audiovisuales de gran calidad, así como la llegada de nuevos formatos como el 4k y el 8k. Comparado con su predecesor H.264/AVC [5], este nuevo estándar es capaz de reducir la tasa de bits hasta un 50 % manteniendo el mismo nivel de calidad subjetiva [14]. No obstante, esta mejora ha venido acompañada de un notable incremento en la complejidad computacional del codificador, llegando a ser impracticable en algunos casos.

En este sentido, el propio estándar incluye las técnicas de paralelización que fueron descritas en la Sección 3.8. Como resultado, se consigue una reducción del tiempo de ejecución a costa de un pequeño coste en términos de eficiencia de codificación. Dado que estas técnicas tienen sus limitaciones, especialmente en la cantidad de paralelismo que pueden introducir, se hace necesario reducir la carga de trabajo que los hilos han de procesar mediante la mejora y optimización de los algoritmos del propio codificador o, como es el caso de este trabajo de investigación, mediante la creación de algoritmos auxiliares que reduzcan la complejidad computacional del resto.

En esta sección, se planteará un algoritmo de pre-análisis para el estándar HEVC que permitirá acelerar, por un lado, algunos módulos del codificador como la predicción inter e intra, y por otro lado, el proceso general de codificación a través de la decisión de modo a nivel de PU o de particionamiento CTU. Hasta el momento, los avances realizados cubren el apartado de la predicción tipo inter. Así, se describirá el proceso de desarrollo y parametrización que ha dado lugar al algoritmo óptimo de pre-análisis para la estimación de movimiento. Igualmente, se mostrarán los resultados obtenidos por éste cuando es comparado con el codificador base. El resto de tareas correspondientes al

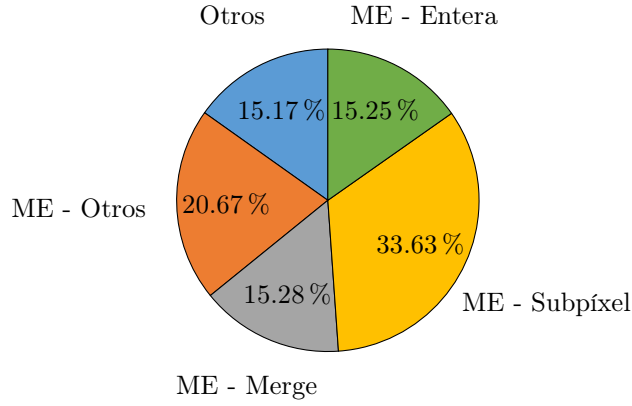


Figura 4.1: Perfil de tiempo para la secuencia Kimono, *Random Access*, QP32

algoritmo de pre-análisis están planificadas para ser desarrolladas a lo largo de la Tesis Doctoral como líneas de trabajo futuras.

4.1. Motivación

Como se ha mencionado previamente, las técnicas de paralelización *Tiles* y WPP sufren de ciertos problemas intrínsecos a su diseño. Por un lado, el primero de ellos supone una notable pérdida en eficiencia de codificación al romper dependencias entre particiones, especialmente cuanto mayor es el número de *tiles* empleado. Por otro lado, WPP sufre de las denominadas *ramping inefficiencies* que limita la aceleración obtenida con este método, ya de por sí acotada por el número de filas de CTUs que puede contener la imagen [16]. En cualquiera de los casos, es necesario disponer de un *hardware* que provea del paralelismo suficiente, lo cual no siempre es posible. Por todas las razones anteriores, numerosos trabajos de la literatura han focalizado sus esfuerzos a mejorar y optimizar los algoritmos incluidos en el estándar.

En nuestro caso, y dado que las técnicas de paralelización tanto en HEVC como en anteriores estándares ya han sido lo suficientemente explotadas, se consideró la posibilidad de actuar sobre la parte del codificador de HEVC para reducir la complejidad computacional del mismo. En este sentido, resulta necesario analizar en primer lugar la distribución de tiempo de los distintos módulos que lo componen para determinar si existe alguno de ellos que sea especialmente costoso en términos de complejidad computacional. La Figura 4.1 muestra el perfil de tiempo de la codificación de la secuencia Kimono (1920×1080, 24 FPS) empleando el modo *Random Access* y un QP de 32. Las condiciones de simulación empleadas son las detalladas posteriormente en la Sección 4.4.2. Como se puede observar en la figura de manera clara, la mayor parte del tiempo de codificación se dedica al proceso de estimación de movimiento (*Motion Estimation*, ME), mientras que tan sólo un 15.17% del tiempo total corresponde al

resto de módulos (etiqueta “otros”). Este último grupo incluye la predicción tipo intra, el proceso de transformada y cuantización, la codificación de entropía y los *in-loop filters*. El proceso de ME, por su parte, incluye las operaciones relacionadas con la predicción tipo inter, incluyendo la búsqueda de vectores de movimiento y la compensación de movimiento (*Motion Compensation*, MC). De esta información se puede extraer principalmente dos conclusiones:

- Gran parte del tiempo de codificación está comprendido por el algoritmo de particionamiento CTU y, a más bajo nivel, por el algoritmo de decisión de modo inter/intra de un PU. Esto implica que una mejora de estos algoritmos supondría una reducción de tiempo en todos los módulos que quedan por debajo, especialmente aquellos relacionados con la predicción.
- El proceso de ME es el que mayor coste computacional supone en todo el proceso de codificación. En concreto, la búsqueda de patrones y su correspondiente refinamiento subpíxel suponen casi la mitad de todo el tiempo total. Esto convierte a este módulo en uno de los mayores focos de atención a la hora de reducir la complejidad del codificador. Hay que tener en cuenta, no obstante, que este porcentaje de tiempo puede variar notablemente en función del patrón de codificación empleado.

Teniendo en cuenta lo anterior, muchas propuestas han ido en la línea de utilizar información de *frames* pasados o de CUs vecinos para acelerar estos módulos. No obstante, esta información no siempre es adecuada para el elemento actual, especialmente en cambios de escena o en la aparición de nuevas texturas en la imagen. Otras propuestas, en cambio, suponen romper dependencias existentes entre CUs o PUs para que puedan ser procesados en paralelo, introduciendo una serie de pérdidas en eficiencia de codificación.

En nuestro caso, proponemos una aproximación distinta al problema. En lugar de actuar en la parte interna del codificador, es posible realizar una fase previa de pre-análisis que realice una versión simplificada de las operaciones llevadas a cabo en éste. De esta manera, se dispondría de una aproximación de la información que será utilizada por el codificador, con lo que se podría acelerar sus operaciones. Así, por ejemplo, se dispondría de una estimación de los vectores de movimiento para la predicción tipo inter, o de la dirección de la interpolación para la predicción tipo intra, con las que, a su vez, podría obtenerse unos costes estimados de ambos tipos de predicción y dar soporte a la decisión de modo, e incluso al particionamiento CTU. En otras palabras, se trata de un algoritmo que daría soporte a múltiples módulos del codificador, tal y como muestra la Figura 4.2, la cual añade en rojo aquellos módulos y relaciones correspondientes al algoritmo de pre-análisis.

Esta aproximación cuenta con una serie de ventajas, entre las que hay que destacar el hecho de que la información obtenida se basa en las imágenes originales de la secuencia,

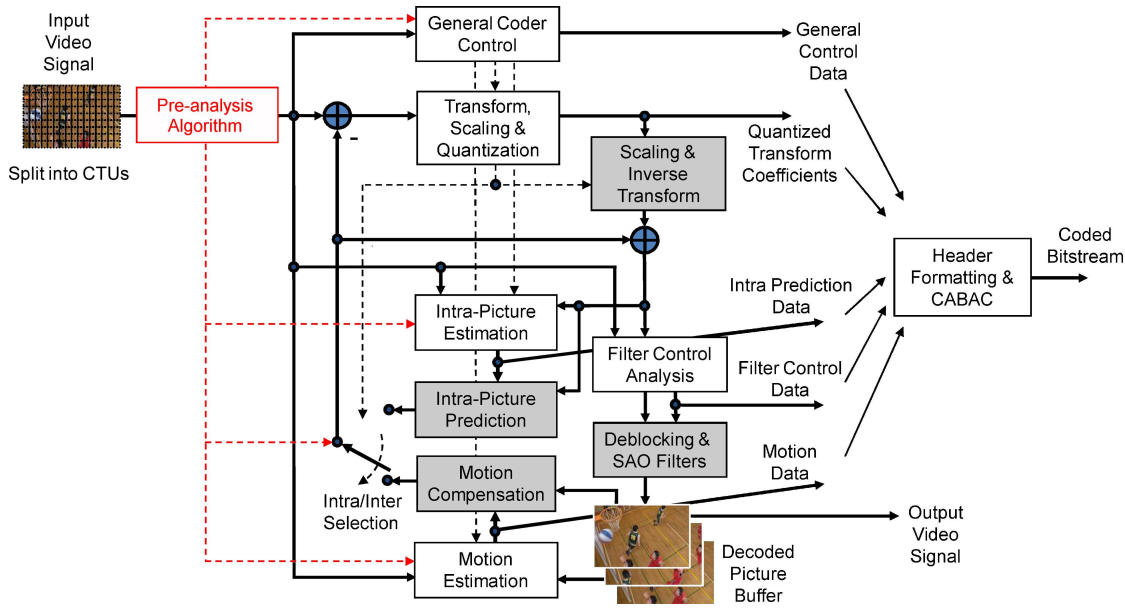


Figura 4.2: Diagrama de bloques de un codificador basado en HEVC empleando pre-análisis (en rojo) [14]

y no en información obtenida a partir de vecinos u otras regiones de la imagen, sin perjuicio de que pueda seguir siendo utilizada si así fuera necesario. Como desventaja, al ser un algoritmo previo a la codificación, podría introducir una latencia inicial antes de poder comenzar con el proceso de codificación. Sin embargo, este tiempo sería mínimo, puesto que correspondería con aquél necesario para preprocesar los primeros *frames*. Además, este hecho es irrelevante en muchos contextos como el de la transcodificación.

4.2. Trabajos relacionados

Esta sección mostrará algunos de los trabajos relacionados con la propuesta descrita en este documento. A pesar de ser un estándar relativamente reciente, ya existen numerosas contribuciones que buscan, en términos generales, reducir la complejidad computacional del codificador, mejorar su eficiencia de codificación o, si resulta posible, ambas opciones a la vez. Todo ello con el objetivo común de no sólo tratar de mejorar las características de los codificadores existentes de HEVC, sino también proporcionar ideas y propuestas para la próxima generación de estándares de codificación de vídeo.

No es posible comenzar sin introducir la propuesta de *C. C. Chi et al.* descrita en [16]. En ella, además de llevar a cabo una comparativa entre *Tiles* y WPP, proponen una mejora de WPP denominada *Overlapped Wavefront* (OWF). Basándose en su funcionamiento, OWF permite superponer el procesamiento de varias imágenes consecutivas de forma concurrente, de manera que cuando un hilo termina de procesar la última fila de CTUs de la imagen actual, puede comenzar a procesar la siguiente. Para

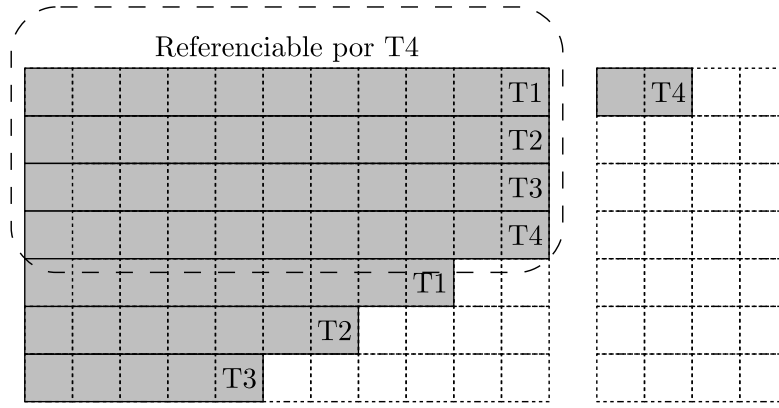


Figura 4.3: Ejemplo de funcionamiento de la estrategia OWF con cuatro hilos

ello, resulta necesario restringir el área de búsqueda establecida sobre los *frames* de referencia que se encuentran todavía en proceso de codificación, tal y como refleja la Figura 4.3. De esta manera, se soluciona el problema de las *ramping inefficiencies* al inicio y al final de cada *frame*. No obstante, su paralelismo puede seguir encontrándose limitado por la cantidad de hilos que es posible lanzar en paralelo en una misma imagen. Por esta razón, posteriormente han surgido otras propuestas como la de *K. Chen et al.* en [17], la cual va un paso más allá y permite que múltiples *frames* puedan ser procesados en paralelo siempre que el área de búsqueda de la imagen de referencia correspondiente se encuentre disponible.

Otras técnicas de paralelización a alto nivel aplicadas a HEVC optan por dividir las imágenes en *slices* independientes, lo que afecta de forma severa a la eficiencia de codificación debido a la ruptura de las vecindades y dependencias existentes [18]. Alternativamente, otros trabajos recurren a la técnica consistente en repartir cada *Group Of Pictures* (GOP) a un elemento de procesamiento distinto, puesto que se trata de conjuntos de imágenes que no guardan dependencias entre sí. Este hecho permite obtener una reducción de tiempo muy significativa [19, 20]. De hecho, *T. K. Heng et al.* consiguen superar la barrera de la codificación en tiempo real de contenidos 4k en [21]. Sin embargo, la desventaja más importante de este método es que puede introducir una latencia inicial que no es permisible en algunos contextos y aplicaciones. Una solución intermedia, por tanto, es la propuesta por *M. Kim et al.* en [22], donde describen un algoritmo capaz de calcular las dependencias entre los *frames* de una secuencia, pudiendo así procesar aquéllos que son independientes entre sí, aunque pertenezcan a un mismo GOP.

En un nivel inferior de abstracción, también existen numerosos trabajos que detallan técnicas de paralelización aplicadas a partes concretas del codificador. Por ejemplo, *C. Yan et al.* plantean en [23] un algoritmo altamente paralelo diseñado para sistemas *manycore* con el objetivo de acelerar el particionamiento CTU. A pesar de alcanzar grandes valores de aceleración, sin embargo, no muestran resultados de eficiencia de codificación. Por su parte, *J. Zhang et al.* aprovechan en [24] la característica de HEVC denominada *Motion Estimation Regions* (MER), la cual permite eliminar las depen-

dencias en la estimación de movimiento de los bloques contiguos dentro de una región. Así, consiguen paralelizar la decisión de modo de forma masiva, si bien obtienen unas pérdidas notables en términos de eficiencia de codificación. Alternativamente, otros trabajos emplean *hardware* especializado como FPGAs o GPUs para acelerar módulos como el de estimación de movimiento [25, 26].

Por otro lado, otro tipo de propuestas dejan a un lado las técnicas de paralelización y adoptan una aproximación distinta, aunque complementaria, consistente en la creación y mejora de algoritmos de codificación de vídeo para reducir la complejidad computacional del codificador. Así, por ejemplo, algunas propuestas optan por reducir la complejidad computacional del esquema de particionamiento CTU mediante técnicas de *early termination* [27] o mediante el análisis de los parámetros espacio-temporales de la secuencia [28]. Otras, en cambio, se centran en la predicción tipo intra, estimando la dirección de predicción a partir de los bloques vecinos [29] o de la propia textura de la imagen [30]. Un tercer grupo de trabajos se encuentra más enfocado a la estimación de movimiento, reduciendo el número de particiones comprobadas en función del movimiento de la imagen [31], empleando umbrales para limitar el uso de diversos modos de predicción tipo inter y sus correspondientes tamaños de CU [32], o calculando información estadística aplicada al movimiento [33]. En la mayor parte de los casos, estas propuestas relacionadas con la predicción tipo inter alcanzan en torno al 50 % de reducción de tiempo con pérdidas asumibles en términos de eficiencia de codificación [34].

En lo que a algoritmos de pre-análisis se refiere, cabe destacar el módulo *look-ahead* incluido en el codificador de H.264/AVC por excelencia, x264 [35]. Este módulo incluye técnicas similares a las que serán descritas en este documento. Su modo de funcionamiento consiste en realizar un submuestreo de las imágenes originales en una proporción 1 : 2 y dividir las en bloques de tamaño 8×8 píxeles. A partir de estos bloques, lleva a cabo tanto una estimación de la dirección de interpolación de la predicción tipo intra, como una pseudoestimación de movimiento para la predicción tipo inter. Como resultado, obtiene, entre otros, unos costes estimados para cada tipo de predicción. Estos costes son usados para acelerar el proceso de codificación, regular el módulo de control de tasa (*Rate Control*, RC) con el que se limita el *bit-rate* de salida y determinar el patrón de codificación más adecuado a utilizar [36]. Sin embargo, resulta imposible encontrar artículos o reportes técnicos que detallen las decisiones de diseño que han dado lugar a dicho módulo. Además, está pensado para ser utilizado con otro estándar, el cual involucra mucha menos complejidad que HEVC.

En cuanto a trabajos relacionados con HEVC se refiere, el algoritmo de codificación en tiempo real introducido previamente implementa una etapa de pre-análisis sobre una GPU con el objetivo de estimar el movimiento de las imágenes [21]. Esto demuestra la importancia de este tipo de algoritmos en codificadores de uso práctico. Por otro lado, *L. Sun et al.* describen en [37] otra etapa de pre-análisis que permite estimar el residuo generado tras la estimación de movimiento. Con esta información, consiguen

ajustar con gran precisión el algoritmo RC del codificador. Su finalidad, por tanto, es distinta a la de este trabajo de investigación. En [38], *T. Mallikarachchi et al.* presentan un algoritmo de análisis de la imagen para localizar zonas homogéneas sobre las que fácilmente se puede determinar el tamaño de las particiones en modos de codificación *All Intra*. El mismo objetivo persiguen *G. Tian* y *S. Goto* en [39] analizando la energía presente en los bloques submuestreados de la imagen.

En definitiva, se puede observar como la comunidad científica está volcada en la tarea de reducir la complejidad computacional del codificador de HEVC a unos niveles asumibles. En esta línea, los algoritmos de pre-análisis representan un papel importante en los codificadores comerciales, principalmente debido a la cantidad de información que pueden aportar al proceso de codificación y a la utilidad que de ella se puede obtener.

4.3. Algoritmo de pre-análisis para la ME

Tal y como se introdujo al inicio de este capítulo, el algoritmo de pre-análisis busca ser una solución integrada para multitud de módulos del codificador de HEVC. En esta sección, será descrita la parte implementada hasta el momento, la cual se corresponde con aquélla destinada al soporte de la estimación de movimiento.

Este algoritmo se divide en dos partes claramente diferenciadas, una de ellas situada en el bloque de pre-análisis, y la otra integrada en el proceso de ME. Estas dos etapas vienen representadas en la Figura 4.4, donde se muestra un esquema simplificado del algoritmo. La etapa de pre-análisis realiza una primera estimación de movimiento del *frame* a codificar sobre un área de búsqueda amplia. Esto proporciona como resultado un mapa de vectores de movimiento aproximados que es utilizado por la segunda etapa, situada en el proceso de ME del codificador. A partir de estos vectores, es posible establecer un punto de partida aproximado para la búsqueda de los PUs a codificar, lo que permite reducir en gran medida el área de búsqueda empleada.

Pese a que pueda parecer que se realiza un doble trabajo al replicar la estimación de movimiento en el bloque de pre-análisis, este hecho no implica que el tiempo total de codificación sea mayor. Por un lado, la estimación de movimiento de la etapa de pre-análisis es muy sencilla y, por tanto, muy rápida. Por otro lado, el tiempo dedicado a la ME se reduce al poder disminuir considerablemente el tamaño del área de búsqueda. Como resultado, la complejidad computacional de la codificación se ve reducida.

A continuación, se explicará más en detalle el funcionamiento de cada una de las partes que componen el algoritmo, así como sus características de diseño.

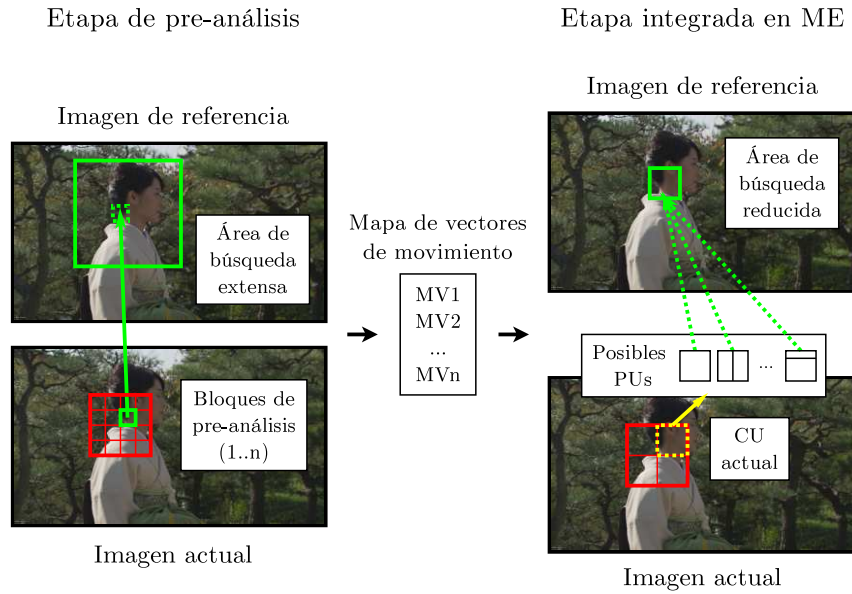


Figura 4.4: Diagrama de bloques de un codificador basado en HEVC empleando pre-análisis (en rojo)

4.3.1. Etapa de pre-análisis

En esta etapa, las imágenes originales de entrada se dividen en bloques cuadrados de igual tamaño. Al igual que si se tratase de PUs, se realiza una estimación de movimiento sobre dichos bloques empleando los *frames* de referencia que corresponden al *frame* procesado. Esta estimación, sin embargo, tiene las siguientes particularidades:

- Dado que el algoritmo procesa una cantidad específica de *frames* previo a su codificación, éste no dispondrá de los *frames* reconstruidos para emplearlos como *frames* de referencia, puesto que no han sido todavía codificados. Por esta razón, ha de emplear las imágenes originales de la secuencia.

En términos generales, las imágenes reconstruidas guardan gran similitud con las originales para valores de QP bajos, mostrando algunas diferencias para valores superiores. No obstante, esto no presenta un problema para el algoritmo, puesto que lo que estamos buscando es simplemente una aproximación de los vectores de movimiento a estimar.

- La estimación de movimiento llevada a cabo se realiza sobre muestras puramente enteras, es decir, no se emplea ningún tipo de refinamiento sub-píxel. Esto ayuda a que esta etapa sea muy poco costosa en tiempo computacional al evitar tener que interpolar las muestras. La justificación de esta decisión de diseño es inmediata si tenemos en cuenta que sólo buscamos una aproximación del vector de movimiento.

Además de estas consideraciones, hay que añadir que el tamaño del área de búsqueda de esta etapa ha de establecerse en un valor lo suficientemente alto para no descartar

los posibles candidatos que se encuentren más alejados del punto inicial de la búsqueda. Reducir este parámetro sería tan perjudicial para la etapa de pre-análisis como lo sería para la ME del codificador en términos de eficiencia de codificación.

Por otro lado, el algoritmo de pre-análisis contenido en esta etapa se ha diseñado de tal manera que sea similar al llevado a cabo en la ME del codificador, replicando en la medida de lo posible las distintas características del mismo. Esto permite que las estimaciones encontradas le sean más útiles al proceso de ME. Estas consideraciones incluyen:

- Emplear el mismo algoritmo de búsqueda y estimación de movimiento que en el módulo ME del codificador. En nuestro caso, se emplea el algoritmo utilizado en los codificadores de referencia de H.264/AVC y HEVC, denominado TZ-Search [40].
- Tomar los vectores de movimiento de los bloques vecinos como predictores de manera similar a como lo hace el algoritmo AMVP en el estándar [14]. En este sentido, se consideran hasta dos predictores, los cuales son tomados de los bloques situados a la izquierda y arriba del bloque actual. En caso de que alguno de ellos no estuviera disponible, se tiene en consideración un candidato temporal, cuyo vector es necesario redimensionar en función de la diferencia temporal entre los *frames*. Si no se hubiera conseguido dos predictores hasta el momento, se completa la lista con vectores nulos. De estos, el que proporciona menor distorsión es el que se utiliza como predictor para la estimación de movimiento.
- Para determinar cuál es el mejor vector de movimiento, se utiliza el conocido modelo lagrangiano de *Rate-Distortion Optimization* (RDO) dado por la siguiente ecuación:

$$J_{MV} = SAD(MV) + \lambda \cdot R(MV)$$

donde J_{MV} representa la función de coste, $SAD(MV)$ es la función de distorsión que se corresponde con la suma de las diferencias absolutas (*Sum of Absolute Differences*, SAD), λ es un multiplicador de Lagrange que depende del QP, y $R(MV)$ representa los bits necesarios para codificar el vector de movimiento.

Como resultado de esta etapa, se genera un mapa de vectores de movimiento que relaciona cada bloque de la imagen actual con un vector por cada *frame* de referencia. Este vector es el mejor encontrado en términos de coste *Rate-Distortion* (RD). La estructura de datos generada es utilizada por la etapa de codificación para acelerar el proceso de ME.

4.3.2. Etapa integrada en la ME

El algoritmo ME del codificador es un proceso complejo que incluye numerosos pasos y operaciones. Estos pasos son los que pueden ser acelerados gracias a la etapa

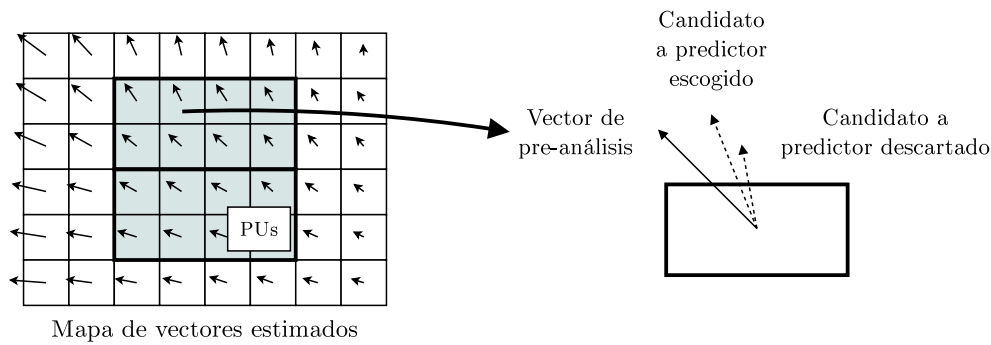


Figura 4.5: Soporte de la etapa de pre-análisis en el algoritmo de elección de predictores

de pre-análisis propuesta, principalmente debido a que los vectores de movimiento pueden ser usados como una primera aproximación de la estimación de movimiento a realizar. Así, los procesos del módulo ME que se ven afectados son dos y serán descritos a continuación.

Por un lado, el algoritmo que escoge los predictores de movimiento basándose en AMVP. Originalmente, este algoritmo comprueba el coste de los dos posibles candidatos definidos por el estándar con el objetivo de seleccionar el que servirá de predictor y, a la vez, de punto de partida para la búsqueda. No obstante, dado que estamos asumiendo que el vector calculado en la etapa de pre-análisis es una buena aproximación, podemos escoger éste como punto inicial de la estimación de movimiento y tomar como predictor aquel candidato que menor cantidad de bits suponga para codificarlo, es decir, el más cercano, tal y como se muestra en la Figura 4.5. Aun así, hay que tener en cuenta lo siguiente:

- En muchos casos, el PU sobre el cual se está realizando la estimación de movimiento no tiene el mismo tamaño que el bloque de pre-análisis, incluso puede ocurrir que tampoco tenga forma cuadrada. En estos casos, si las dimensiones del PU son menores a las del bloque, la solución adoptada es tomar el vector de movimiento estimado directamente. Sin embargo, en caso de no ser así, se toma la mediana (en términos de distancia euclídea) de entre todos los vectores correspondientes a los bloques que ocupa el PU.
- Puede suceder que la estimación obtenida en la etapa de pre-análisis no sea del todo adecuada para el PU a codificar. Esto puede darse cuando el vector de pre-análisis sea muy diferente a los predictores. Por ello, si la distancia del vector al predictor es superior al 90 % del rango de búsqueda o a un máximo de 5 píxeles, el coste de tanto los candidatos como del vector de pre-análisis es comprobado, puesto que a una distancia menor el propio algoritmo de búsqueda podría discriminar entre ambos [40]. Así, si el coste correspondiente al vector de pre-análisis es el menor de todos, se utiliza como punto inicial de la búsqueda como se indicó previamente, y si no, el vector se ignora completamente.

Por otro lado, el proceso de estimación de movimiento del módulo de ME también es acelerado gracias a la selección más acertada del punto inicial de la búsqueda. Esto permite que el área de referencia pueda ser notablemente reducida, al poder asumir que el vector de movimiento del PU actual se encuentra cercano a dicho punto inicial.

En definitiva, la etapa de pre-análisis permite reducir el número de posiciones comprobadas en el área de referencia y, en segundo término, reducir el número de operaciones SAD llevadas a cabo. Como resultado, el tiempo empleado en el módulo ME se ve reducido, especialmente en lo que se refiere a la búsqueda entera, lo cual influye de manera directa en el tiempo total de codificación.

4.4. Configuración del algoritmo de pre-análisis para la ME

La sección anterior ha detallado el funcionamiento del algoritmo de pre-análisis para el módulo ME. No obstante, todavía no se ha especificado el valor de los parámetros que definen completamente su comportamiento, los cuales incluyen principalmente el tamaño del bloque de pre-análisis y el tamaño del área de búsqueda de la ME. El valor de estos parámetros se calculará de manera experimental, analizando las ventajas y desventajas asociados a cada uno de ellos.

Esta sección mostrará las métricas empleadas, detallará la plataforma *hardware* y *software* utilizada en el análisis experimental, establecerá la línea base que se empleará como referencia en las comparaciones y, finalmente, estudiará los parámetros indicados previamente.

4.4.1. Métricas empleadas

Para poder analizar el desempeño de las distintas propuestas es necesario poder medir dos parámetros básicos: el tiempo que ahorran y su eficiencia de codificación. Para ello, se empleará tanto el *speed-up* como la reducción de tiempo por un lado, mientras que por otro se considerará la métrica denominada *Bjontegaard Delta Rate* (BD-rate) [41].

El *speed-up* permite medir la mejora relativa de rendimiento entre dos propuestas. Su valor se calcula a partir de la Ecuación 4.1,

$$SU = \frac{T_{base}}{T_{propuesta}} \quad (4.1)$$

donde T_{base} se corresponde con el tiempo de la versión base y $T_{propuesta}$ el de la propuesta que se desea evaluar. Así, un valor superior a la unidad representa la proporción de mejora alcanzada.

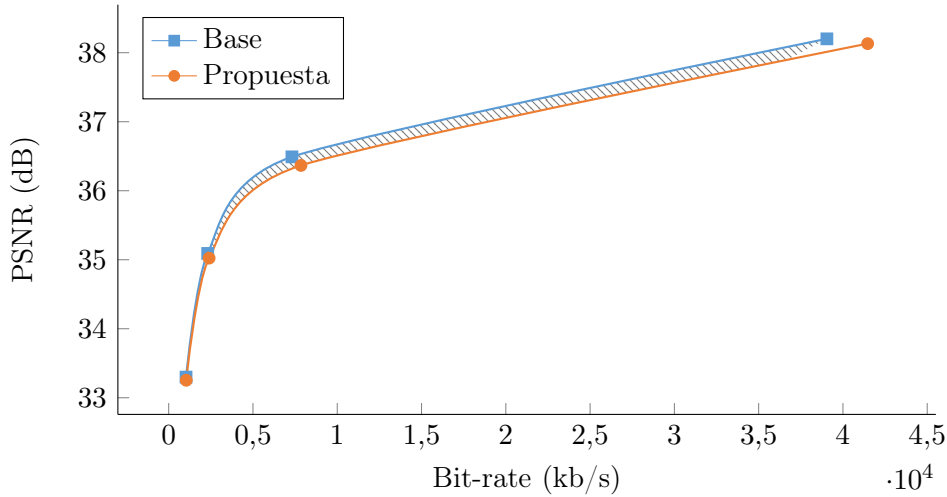


Figura 4.6: Ejemplo del cálculo de BD-rate sobre unas curvas de bit-rate/PSNR

Por su parte, la métrica de reducción de tiempo es similar a la anterior, solo que desde un punto de vista diferente. En su lugar, mide el porcentaje de tiempo que una propuesta ahorra con respecto a su versión original, y viene dada por la Ecuación 4.2,

$$TR = \frac{T_{base} - T_{propuesta}}{T_{base}} = 1 - \frac{1}{SU} \quad (4.2)$$

donde T_{base} y $T_{propuesta}$ representan lo mismo que en el caso anterior, y SU es el *speed-up* de una propuesta.

Por último, la eficiencia de codificación ha sido tradicionalmente obtenida a partir de dos parámetros. Por un lado, la calidad objetiva de la imagen, para lo cual generalmente se emplea la relación *Peak Signal to Noise Ratio* (PSNR) que tiene en cuenta el error cuadrático medio y, por otro lado, la tasa de bits para medir la capacidad de compresión del codificador. De esta manera, se consigue generar una curva *bit-rate*/PSNR como la mostrada en la Figura 4.6, donde cada punto se obtiene a partir de la codificación de la misma secuencia con QP distinto.

En un esfuerzo por aunar los resultados y hacerlos más comparables entre propuestas, se creó la métrica BD-rate [41], la cual será la utilizada en estos experimentos. Esta métrica asimila en un único valor tanto la calidad visual de una secuencia como su *bit-rate*, de manera que representa el incremento (o decremento) del *bit-rate* entre dos secuencias con la misma calidad objetiva. En el ejemplo de la Figura 4.6 previamente mencionado, el BD-rate obtenido por la propuesta viene representado por el área contenida entre ambas curvas, el cual se corresponde con el 12.53 %.

4.4.2. Configuración experimental

Los experimentos han sido ejecutados sobre una versión optimizada del software de referencia del estándar: HEVC Test Model (HM-14.0) [9]. En esta versión, algunas

herramientas de codificación han sido deshabilitadas, buscando reducir la complejidad computacional del codificador sin penalizar en exceso la eficiencia de codificación. Entre ellas, se encuentra el particionado RQT, el *skip* de la transformada y algunos modos intra para *chroma*. En relación a los esquemas *full-RD*, no son considerados los candidatos *merge* y se reduce el número de candidatos para la intrapredicción a 3. Además, el algoritmo de búsqueda de la estimación de movimiento ha sido modificado para obviar el paso en que se realiza un barrido tipo *raster* por el área de referencia [40]. Por último, numerosas operaciones han sido optimizadas empleando el paradigma *Single Instruction Multiple Data* (SIMD) con el conjunto de instrucciones Intel AVX2, consiguiendo una notable aceleración a coste nulo en términos de eficiencia de codificación. En media, todas estas modificaciones proveen un $4.02\times$ de aceleración a un coste de tan sólo un incremento del 3.5% en BD-rate en comparación con la versión base de HM.

En relación a la configuración utilizada en la codificación, se ha empleado el patrón *Random Access* (ver Sección 3.10), puesto que es el más extendido en todo el conjunto de contenidos multimedia existente. Salvo las optimizaciones antes mencionadas, los parámetros de codificación empleados son los especificados en el documento emitido por la JCT-VC para establecer un marco común de pruebas [42], entre los que se incluyen:

- Las secuencias han sido codificadas con los valores de QP 22, 27, 32 y 37.
- Se ha utilizado el formato chroma 4:2:0 y una profundidad de 8 bits (perfil *Main*).
- Se han codificado secuencias de clase A hasta la D, las cuales incluyen las resoluciones 2560×1600 (A), 1920×1080 (B), 832×480 (C) y 416×240 (D).

Además, el rango de búsqueda para la etapa de pre-análisis se ha establecido al valor por defecto correspondiente a la estimación de movimiento del módulo ME en el codificador original, es decir, 64. Esto quiere decir que el tamaño de la región de búsqueda de pre-análisis para todas las pruebas es de 128×128 , lo cual no tiene por qué suponer que en todos los casos se examine toda su extensión. Tal y como se indicó en la Sección 4.3.1, el reducir este parámetro a un valor demasiado bajo podría afectar gravemente a la eficiencia de codificación.

La plataforma *hardware* sobre la que se han ejecutado las codificaciones está compuesta por procesadores Intel Core i7-4770T a una frecuencia de 2.50 GHz y 12 Gb de memoria principal. Las tecnologías Turbo Boost [43] e Hyperthreading [44] del procesador han sido deshabilitadas para permitir la reproducibilidad de los resultados. El *encoder* se ha compilado con GCC 4.8.2-19 y se ha ejecutado sobre Ubuntu 14.04.

Cuadro 4.1: Resultados de BD-rate para diferentes tamaño de bloque de pre-análisis

		BD-rate (%)			
		8×8	16×16	32×32	64×64
A	Traffic	0.68	0.73	1.39	3.14
	PeopleOnStreet	1.31	1.29	1.61	2.61
	NebutaFestival	0.07	0.16	0.05	0.00
	SteamLocomotive	2.29	0.22	-0.22	-0.56
B	Kimono	0.63	0.37	0.44	0.86
	ParkScene	0.40	0.48	0.69	1.08
	Cactus	0.77	0.58	0.70	1.38
	BasketballDrive	0.79	0.50	0.41	0.66
	BQTerrace	0.49	0.51	0.80	1.50
C	BasketballDrill	0.88	0.75	1.28	2.90
	BQMall	0.43	0.49	0.82	1.41
	PartyScene	0.51	0.67	0.86	1.20
	RaceHorses	1.50	1.69	2.25	3.60
D	BasketballPass	0.63	0.56	0.86	1.64
	BQSquare	0.12	0.26	0.44	0.49
	BlowingBubbles	0.34	0.61	0.92	1.35
	RaceHorses	1.17	1.38	1.70	2.92
Media		0.76	0.66	0.88	1.53

4.4.3. Tamaño del bloque de pre-análisis

El tamaño del bloque de pre-análisis determina la granularidad de la estimación de movimiento llevada a cabo en esta etapa. Un tamaño de bloque más pequeño proporciona como resultado un mayor número de vectores por *frame* y, por tanto, incrementa la complejidad del algoritmo. Por el contrario, un tamaño de bloque demasiado grande no es capaz de capturar el suficiente detalle del movimiento en una escena.

Para determinar el tamaño de bloque de pre-análisis óptimo, se han llevado a cabo experimentos para tamaños de bloque de 8×8 , 16×16 , 32×32 y 64×64 . No obstante, no tiene sentido que el rango de búsqueda del módulo ME se mantenga a su valor por defecto, el cual es relativamente alto y equivalente al establecido para la etapa de pre-análisis, puesto que no es el objetivo del algoritmo propuesto. Por ello, para representar un escenario práctico del algoritmo de pre-análisis, el rango de búsqueda para la estimación de movimiento se ha establecido a 4 para estos experimentos. Cabe mencionar, sin embargo, que se han realizado pruebas con otros rangos de búsqueda y el comportamiento de los resultados es similar.

La Tabla 4.1 muestra los resultados obtenidos en cada secuencia, los cuales quedan representados y agrupados por clases en la Figura 4.7. Las conclusiones que es posible extraer de ellos son muy diversas. En primer lugar, los resultados indican que los bloques de 8×8 y 16×16 son aquéllos que obtienen los mejores resultados en términos generales. Mención especial merece la secuencia SteamLocomotive, en que el tamaño de bloque 8×8 obtiene una notable penalización en términos de eficiencia de codificación. Esto se

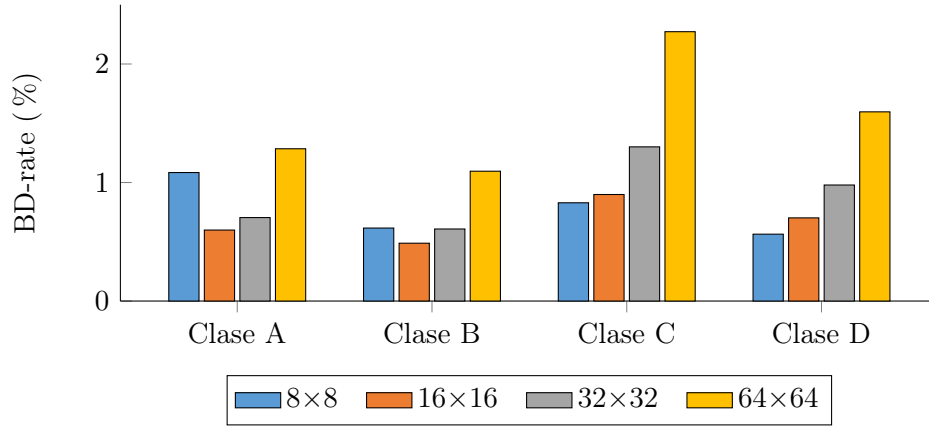


Figura 4.7: Resultados de BD-rate para diferentes tamaño de bloque de pre-análisis

debe a que esta secuencia muestra un tren de vapor que crea una textura uniforme y homogénea por gran parte de la imagen. Por ello, un tamaño de bloque tan pequeño no logra capturar correctamente el movimiento para PUs de mayor tamaño, incurriendo en un mayor BD-rate.

En términos de complejidad computacional, el tiempo destinado a la etapa de pre-análisis es mayor cuanto menor es el tamaño de bloque, en una relación aproximada de hasta el doble por cada reducción. Sin embargo, esta complejidad añadida es casi despreciable cuando se compara con el tiempo total de codificación. Por este motivo, no resulta de utilidad mostrar los datos de aceleración en este apartado.

Teniendo en cuenta todo lo anterior, el tamaño de bloque óptimo para el algoritmo de pre-análisis es 16×16 , y es el que se utilizará en el resto de experimentos. Esta decisión radica en el hecho de que en media obtiene los mejores resultados, y en que el tamaño de bloque 8×8 , el cual obtiene también buenos resultados en general, puede presentar problemas en secuencias de alta resolución con detalles u objetos de gran tamaño.

4.4.4. Rango de búsqueda del módulo ME

Tal y como fue descrito previamente, el algoritmo de pre-análisis es capaz de proveer una estimación de los vectores de movimiento que serán utilizados en el módulo ME. Por ello, pueden ser utilizados como un buen punto de partida para la estimación de movimiento realizada en dicho módulo. De este modo, se puede asumir que no es necesario emplear una gran área de búsqueda para encontrar el candidato óptimo. Este apartado, por tanto, tendrá como objetivo determinar el rango de búsqueda que permita obtener como resultado un compromiso entre eficiencia de codificación y disminución de la complejidad computacional del codificador.

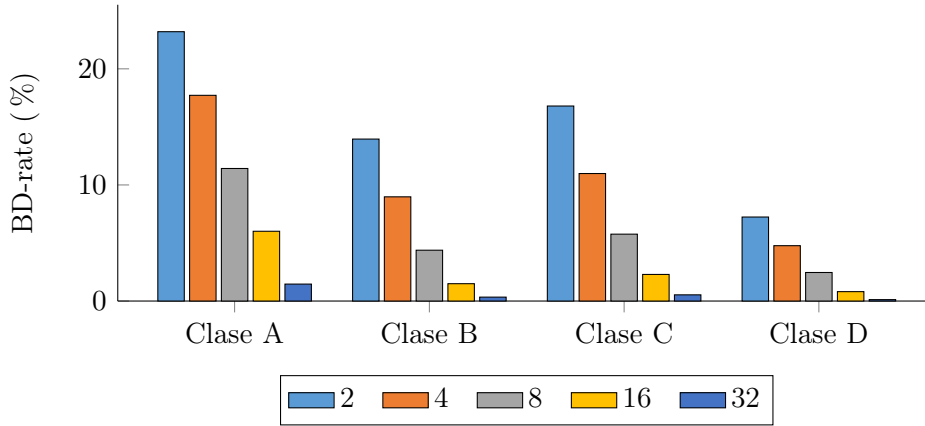


Figura 4.8: Resultados de BD-rate para diferentes rangos de búsqueda sin el algoritmo de pre-análisis

Como primer punto de comparación, la Figura 4.8 muestra los resultados de eficiencia de codificación obtenidos a partir del codificador optimizado base, es decir, sin algoritmo de pre-análisis, cuando el rango de búsqueda es menor a su valor por defecto, 64. Como se puede observar, las pérdidas en BD-rate causadas por esta reducción son impracticables, llegando hasta más de un 20 % en algunos casos. Además, esta falta de precisión en la estimación de movimiento hace que otros módulos tengan que llevar a cabo un esfuerzo mayor en sus tareas, por lo que la aceleración total conseguida se encuentra limitada a $1.06\times$, lo que equivale a una reducción del 5.61 % del tiempo total.

En contraposición, al utilizar el algoritmo de pre-análisis sobre el codificador con un tamaño de bloque de 16×16 , los resultados de BD-rate y complejidad computacional correspondientes al rango de búsqueda del módulo ME obtenidos son los mostrados en las Tablas 4.2 y 4.3, respectivamente. Estos mismos resultados vienen representados agrupados por clases en la Figura 4.9. Como se puede observar, estos resultados son mucho mejores que los presentados previamente. La eficiencia de codificación aumenta conforme así lo hace el rango de búsqueda, en términos generales, mostrando la aceleración el comportamiento justamente inverso. Los casos excepcionales en que esto no es así se debe a que el algoritmo realiza decisiones que dan lugar a un resultado subóptimo.

En vista de lo anterior, queda claro que es necesario buscar un compromiso entre el rango de búsqueda utilizado y los resultados obtenidos. La conclusión que se puede extraer de este apartado, por tanto, es que es posible reducir notablemente el rango de búsqueda del módulo ME sin que la eficiencia de codificación se vea gravemente afectada, gracias al uso del algoritmo de pre-análisis.

Cuadro 4.2: Resultados de BD-rate para diferentes rangos de búsqueda del módulo ME aplicando el algoritmo de pre-análisis

		BD-rate (%)					
		2	4	8	16	32	64
A	Traffic	0.75	0.73	0.58	0.43	0.27	0.14
	PeopleOnStreet	1.61	1.29	0.99	0.75	0.62	0.59
	NebutaFestival	0.02	0.16	-0.07	0.04	-0.02	0.03
	SteamLocomotive	0.31	0.22	0.21	-0.10	-0.57	-0.90
B	Kimono	0.46	0.37	0.29	0.14	0.07	0.00
	ParkScene	0.62	0.48	0.31	0.19	0.11	0.10
	Cactus	0.66	0.58	0.48	0.41	0.35	0.32
	BasketballDrive	0.63	0.50	0.41	0.29	0.20	0.17
	BQTerrace	0.48	0.51	0.41	0.32	0.25	0.18
C	BasketballDrill	0.91	0.75	0.66	0.51	0.31	0.27
	BQMall	0.70	0.49	0.33	0.22	0.18	0.17
	PartyScene	0.89	0.67	0.43	0.29	0.26	0.26
	RaceHorses	2.23	1.69	1.05	0.49	0.21	0.11
D	BasketballPass	0.95	0.56	0.41	0.23	0.16	0.05
	BQSquare	0.32	0.26	0.25	0.08	0.12	0.10
	BlowingBubbles	0.86	0.61	0.46	0.33	0.24	0.26
	RaceHorses	1.88	1.38	0.84	0.57	0.29	0.19
Media		0.84	0.66	0.47	0.31	0.18	0.12

Cuadro 4.3: Resultados de tiempo para diferentes rangos de búsqueda del módulo ME aplicando el algoritmo de pre-análisis

		Speed-up						Reducción de tiempo (%)					
		2	4	8	16	32	64	2	4	8	16	32	64
A	Traffic	1.13	1.11	1.11	1.10	1.10	1.10	11.25	10.24	10.14	9.24	9.44	9.03
	PeopleOnStreet	1.16	1.13	1.13	1.10	1.09	1.07	13.64	11.28	11.39	9.30	8.21	6.91
	NebutaFestival	1.14	1.13	1.12	1.11	1.10	1.08	12.40	11.30	10.81	9.88	9.18	7.22
	SteamLocomotive	1.14	1.13	1.11	1.09	1.07	1.06	12.17	11.44	9.71	8.61	6.27	5.86
B	Kimono	1.15	1.13	1.12	1.10	1.08	1.08	13.41	11.36	10.63	9.31	7.74	7.13
	ParkScene	1.14	1.12	1.12	1.11	1.10	1.10	11.99	10.94	10.31	9.72	8.95	8.74
	Cactus	1.14	1.12	1.11	1.11	1.09	1.08	11.93	10.89	9.76	9.53	8.56	7.40
	BasketballDrive	1.16	1.14	1.13	1.09	1.09	1.07	13.98	12.56	11.42	8.57	7.93	6.48
	BQTerrace	1.12	1.11	1.11	1.10	1.08	1.09	10.98	10.09	9.59	9.26	7.33	8.09
C	BasketballDrill	1.14	1.12	1.12	1.10	1.09	1.08	12.51	10.72	10.83	9.05	8.60	7.56
	BQMall	1.16	1.12	1.14	1.11	1.11	1.09	13.44	10.52	12.47	10.08	10.02	8.62
	PartyScene	1.15	1.12	1.10	1.11	1.10	1.10	12.83	10.31	9.38	10.31	9.31	9.02
	RaceHorses	1.18	1.17	1.14	1.13	1.10	1.07	15.25	14.47	12.38	11.28	9.46	6.16
D	BasketballPass	1.19	1.17	1.14	1.14	1.10	1.08	15.70	14.46	12.31	12.03	9.35	7.09
	BQSquare	1.12	1.11	1.10	1.12	1.10	1.10	10.99	9.50	8.85	10.63	9.14	8.76
	BlowingBubbles	1.12	1.10	1.12	1.09	1.09	1.08	10.55	9.01	11.02	7.91	8.50	7.26
	RaceHorses	1.15	1.14	1.17	1.13	1.11	1.11	13.10	12.13	14.26	11.28	10.16	9.86
Media		1.15	1.13	1.12	1.11	1.10	1.08	12.71	11.25	10.90	9.76	8.72	7.72

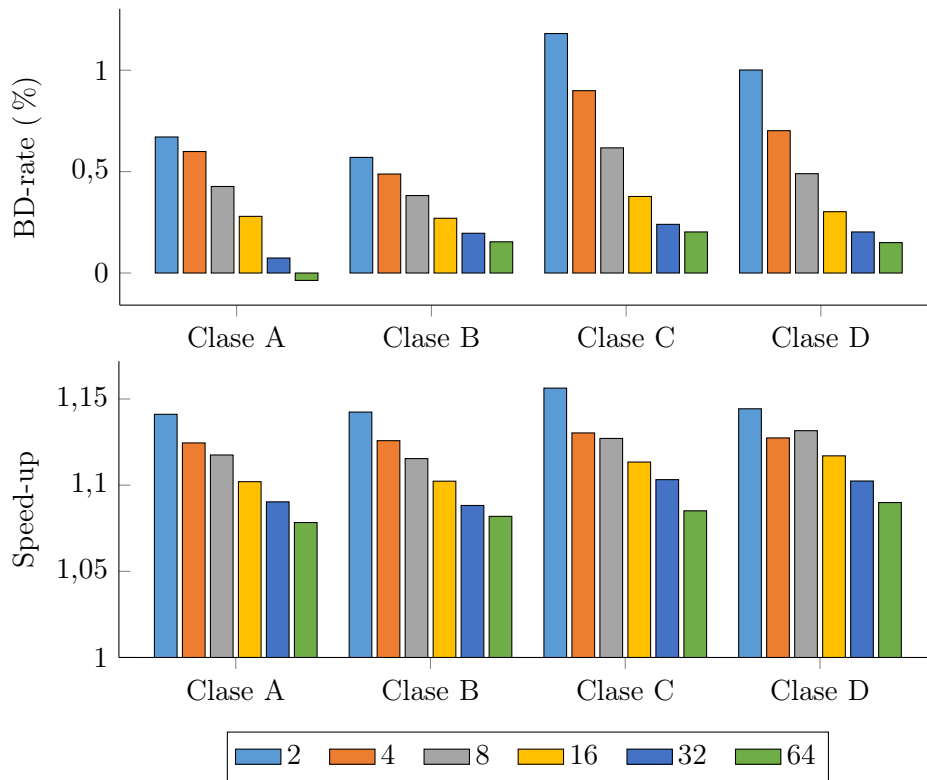


Figura 4.9: Resultados de BD-rate y tiempo para diferentes rangos de búsqueda del módulo ME aplicando el algoritmo de pre-análisis

4.4.5. Sustitución del algoritmo de búsqueda en el módulo ME

Las anteriores secciones analizaron y definieron algunos de los parámetros que están relacionados con el algoritmo de pre-análisis. En este punto, es necesario tener en cuenta que el algoritmo de búsqueda implementado en el codificador está diseñado para llevar a cabo su operación sobre áreas de búsqueda de gran extensión. Por ello, a raíz de la conclusión extraída en el apartado anterior, es fácil pensar que un algoritmo de búsqueda local sería más adecuado para rangos de búsqueda menores, pudiendo obtener los mismos o mejores resultados que el algoritmo por defecto y en un tiempo menor.

El algoritmo original del codificador HM funciona de la siguiente manera:

1. En primer lugar, comprueba cuál es el mejor punto de partida, si el predictor o el vector nulo.
2. A continuación, realiza una búsqueda en forma de diamante, tal y como se representa en la Figura 4.10a. El algoritmo está diseñado para detener esta primera búsqueda si después de tres rondas desde el último mejor candidato no se encuentra ningún otro que mejore la eficiencia de codificación.

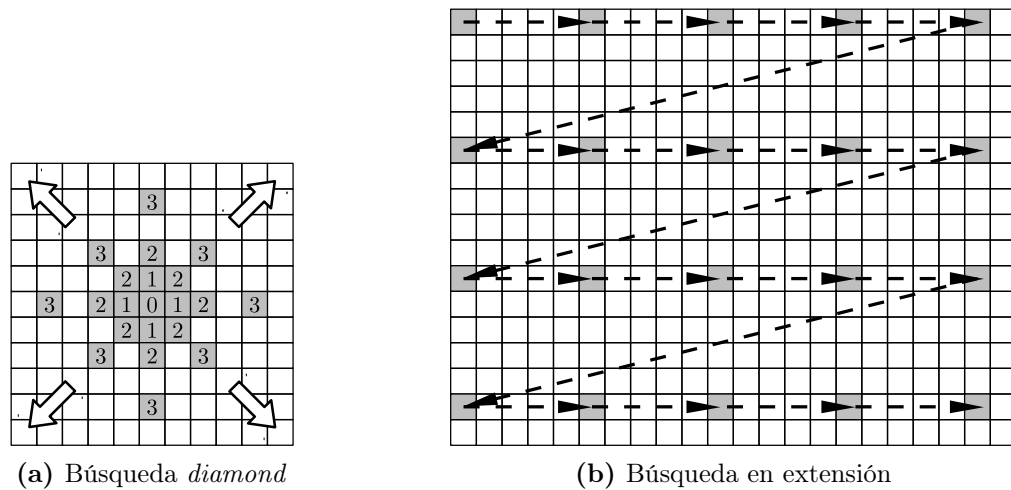


Figura 4.10: Pasos ejecutados en el algoritmo de búsqueda por defecto

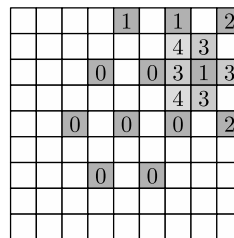


Figura 4.11: Algoritmo de búsqueda propuesto con patrón hexagonal

3. En caso de que la distancia al mejor candidato encontrado desde el punto inicial sea mayor que cinco píxeles, se realiza una búsqueda por todo el área de referencia a intervalos de cinco muestras (*raster scan*), tal y como muestra la Figura 4.10b.
4. Por último, tanto si ha realizado el refinamiento anterior o no, se vuelve a realizar una búsqueda diamante tomando como punto inicial el mejor candidato de los pasos anteriores.

Cabe añadir que, tal y como se especificó en la Sección 4.4.2 de configuración experimental, la versión optimizada del codificador dispone de una versión modificada de este algoritmo. En ella, el paso correspondiente al *raster scan* ha sido eliminado, puesto que introducía un trabajo adicional al algoritmo sin obtener un beneficio claro a partir de él.

En el caso en el que nos encontramos, donde el área de búsqueda es lo suficientemente reducida, el algoritmo anteriormente descrito llevaría a cabo dos búsquedas diamante sobre prácticamente la totalidad del área e incluso repitiendo posiciones. Este problema puede verse resuelto aplicando otro tipo de búsquedas. En nuestro caso, se utilizará uno de los patrones más empleados en algoritmos de búsqueda: el hexagonal. Un ejemplo del algoritmo propuesto viene reflejado en la Figura 4.11:

1. Primeramente, se realiza una búsqueda empleando el patrón hexagonal en el sentido que indique el mejor candidato en todo momento (pasos 0, 1 y 2 en la

Cuadro 4.4: Resultados de BD-rate para diferentes rangos de búsqueda aplicando el algoritmo de pre-análisis y el nuevo algoritmo de búsqueda

		BD-rate (%)		
		2	4	8
A	Traffic	0.88	0.87	0.85
	PeopleOnStreet	1.93	1.82	1.76
	NebutaFestival	0.01	-0.03	0.10
	SteamLocomotive	0.33	0.26	0.05
B	Kimono	0.55	0.45	0.36
	ParkScene	0.67	0.60	0.53
	Cactus	0.73	0.73	0.68
	BasketballDrive	0.72	0.71	0.64
	BQTerrace	0.50	0.55	0.55
C	BasketballDrill	0.95	0.88	0.85
	BQMall	0.82	0.77	0.74
	PartyScene	0.99	0.90	0.86
	RaceHorses	2.44	2.22	2.04
D	BasketballPass	1.08	1.01	0.87
	BQSquare	0.43	0.40	0.35
	BlowingBubbles	0.96	0.86	0.85
	RaceHorses	2.13	1.88	1.83
Media		0.95	0.87	0.82

figura). Esta búsqueda se detiene cuando no se encuentra ningún candidato mejor al actual.

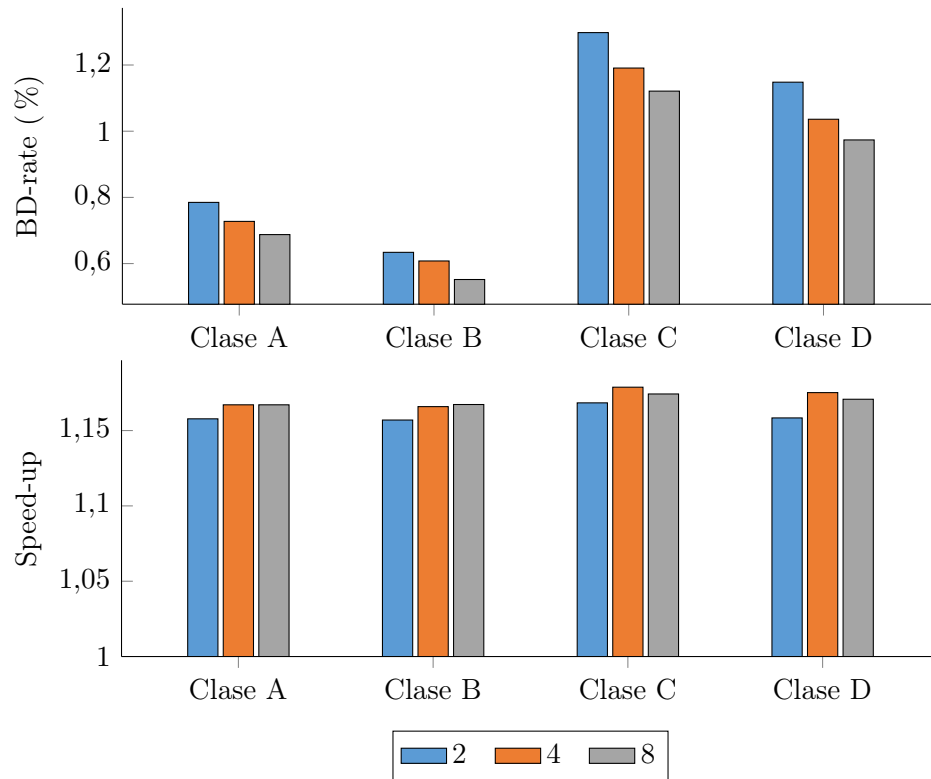
2. Se comprueban las posiciones laterales, superior e inferior del candidato encontrado en la etapa anterior (paso 3 en la figura).
3. En caso de que se encuentre un nuevo mejor candidato, se comprueban las posiciones situadas inmediatamente a los lados de éste (paso 4 de la figura).

Como se puede observar, este algoritmo de búsqueda trata de encontrar un mínimo local en todo el área de búsqueda sin repetir posiciones. De esta manera, el número de operaciones realizadas en total es mucho menor.

Las Tablas 4.4 y 4.5 muestran los resultados de BD-rate y tiempo, respectivamente, obtenidos por el codificador aplicando el algoritmo de pre-análisis y el algoritmo de búsqueda definido en este apartado. Dado que estos algoritmos en su conjunto están pensados para rangos de búsqueda menores, los experimentos se han limitado al subconjunto formado por 2, 4 y 8. La Figura 4.12 contiene agrupados por clase los resultados mostrados en las tablas. Si comparamos estos resultados con los del apartado anterior, podremos identificar dos consecuencias del algoritmo propuesto: por un lado, que la aceleración conseguida es mayor, si bien la eficiencia de codificación se ve levemente afectada y, por otro lado, que al tratarse de un método de búsqueda local, aumentar el rango de búsqueda no influye notablemente en el resultado.

Cuadro 4.5: Resultados de tiempo para diferentes rangos de búsqueda aplicando el algoritmo de pre-análisis y el nuevo algoritmo de búsqueda

		Speed-up			Reducción de tiempo (%)		
		2	4	8	2	4	8
A	Traffic	1.15	1.15	1.15	12.70	13.13	12.92
	PeopleOnStreet	1.17	1.18	1.18	14.70	14.97	15.15
	NebutaFestival	1.15	1.17	1.18	12.86	14.20	14.92
	SteamLocomotive	1.17	1.18	1.17	14.24	14.96	14.26
B	Kimono	1.17	1.17	1.18	14.32	14.74	14.91
	ParkScene	1.15	1.16	1.16	13.18	13.82	14.05
	Cactus	1.15	1.16	1.16	13.01	13.75	13.83
	BasketballDrive	1.18	1.19	1.19	15.15	15.93	15.87
	BQTerrace	1.14	1.15	1.15	12.18	12.85	12.95
C	BasketballDrill	1.16	1.16	1.16	13.45	13.92	13.91
	BQMall	1.17	1.18	1.17	14.21	15.10	14.62
	PartyScene	1.15	1.16	1.16	13.31	13.67	13.79
	RaceHorses	1.20	1.22	1.21	16.62	17.91	17.01
D	BasketballPass	1.19	1.21	1.21	15.62	17.02	17.23
	BQSquare	1.13	1.14	1.13	11.87	12.08	11.84
	BlowingBubbles	1.13	1.15	1.14	11.17	13.12	12.65
	RaceHorses	1.19	1.21	1.20	15.92	17.29	16.49
Media		1.16	1.17	1.17	13.80	14.62	14.49

**Figura 4.12:** Resultados de BD-rate y tiempo para diferentes rangos de búsqueda aplicando el algoritmo de pre-análisis y el nuevo algoritmo de búsqueda

Como era de esperar, la eficiencia de codificación aumenta conforme así lo hace el rango de búsqueda. No obstante, esto no quiere decir que la aceleración siga un comportamiento inversamente proporcional. De hecho, la aceleración conseguida para un rango de 2 es menor que la de otras alternativas debido a que los errores de predicción suponen un mayor trabajo a otros módulos del codificador.

En relación a los rangos de búsqueda 4 y 8, ambos obtienen similares resultados de eficiencia de codificación y de aceleración. En nuestro caso, optaremos por escoger un rango de 4 píxeles por obtener una aceleración media algo superior, y por reducir en mayor medida el número de operaciones realizadas en la estimación de movimiento.

4.5. Resumen de resultados

A lo largo de toda la sección anterior, se ha buscado la configuración óptima del algoritmo de pre-análisis propuesto. Esta búsqueda se ha llevado a cabo paso a paso y teniendo en cuenta los resultados de los apartados previos. La configuración encontrada incluye los siguientes parámetros:

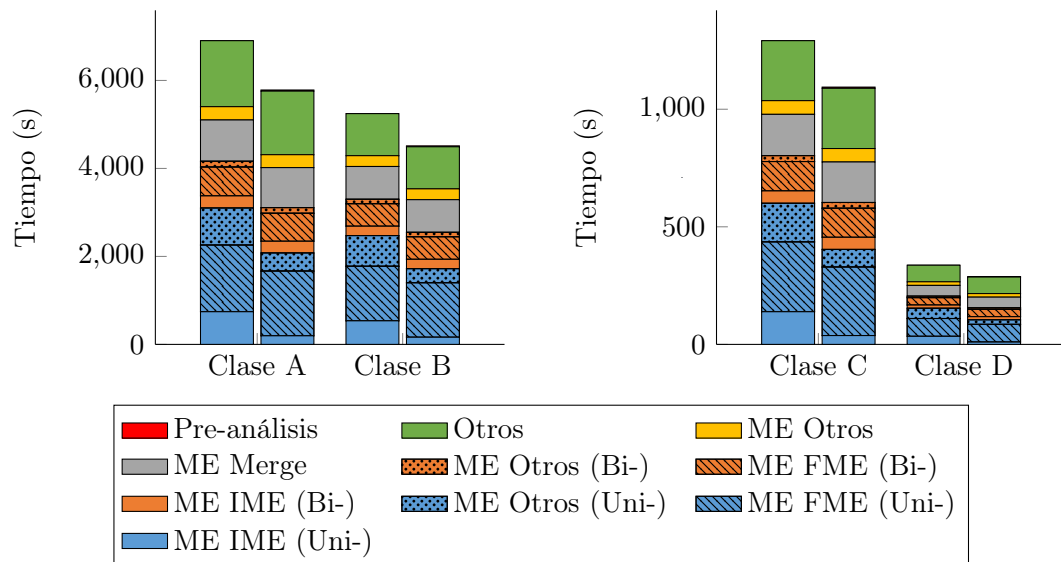
- Un rango de búsqueda de la etapa de pre-análisis de 64 píxeles, lo que equivale a un área de 128×128 .
- Un tamaño de bloque de pre-análisis de 16×16 píxeles.
- Un rango de búsqueda de la etapa del módulo ME de 4 píxeles, lo que equivale a un área de 8×8 .

Teniendo en cuenta esta configuración, la Tabla 4.6 muestra detalladamente los resultados obtenidos por el codificador empleando el algoritmo de pre-análisis propuesto. Además de la eficiencia y el tiempo de codificación que se ha mostrado en otros apartados, también se incluye la aceleración correspondiente a la estimación de movimiento entera unidireccional del módulo ME, la cual se ve afectada por el algoritmo propuesto. En este sentido, también se indica el porcentaje de operaciones SAD que se ha ahorrado en este punto de la codificación.

Los resultados muestran que el algoritmo propuesto es capaz de acelerar la codificación en un factor $1.17 \times$, o dicho de otro modo, reduce un 14.62 % el tiempo total de codificación a costa de una penalización del 0.87 % en BD-rate. Este ahorro en tiempo se produce por la reducción del tamaño del área de búsqueda en el módulo ME y por la sustitución del algoritmo de búsqueda empleado, lo cual permite tener una aceleración de $3.55 \times$ de la estimación de movimiento entera unidireccional, es decir, una reducción del 70.37 % del tiempo dedicado a esta etapa. Como causa subyacente, se encuentra

Cuadro 4.6: Resultados del algoritmo de pre-análisis empleando la configuración óptima propuesta

		BD-rate (%)	Tiempo de codificación		Tiempo ME entera		Reducción SAD (%)
			Speed-up	Reducción (%)	Speed-up	Reducción (%)	
A	Traffic	0.87	1.15	13.13	2.43	58.84	65.84
	PeopleOnStreet	1.82	1.18	14.97	4.19	76.14	83.50
	NebutaFestival	−0.03	1.17	14.20	4.09	75.53	80.28
	SteamLocomotive	0.26	1.18	14.96	3.79	73.65	81.81
B	Kimono	0.45	1.17	14.74	3.63	72.43	80.93
	ParkScene	0.60	1.16	13.82	2.82	64.48	72.08
	Cactus	0.73	1.16	13.75	3.18	68.51	75.99
	BasketballDrive	0.71	1.19	15.93	4.15	75.88	83.76
	BQTerrace	0.55	1.15	12.85	2.61	61.64	69.34
C	BasketballDrill	0.88	1.16	13.92	3.64	72.54	79.72
	BQMall	0.77	1.18	15.10	3.37	70.33	76.31
	PartyScene	0.90	1.16	13.67	3.10	67.78	73.15
	RaceHorses	2.22	1.22	17.91	5.21	80.82	86.69
D	BasketballPass	1.01	1.21	17.02	4.44	77.50	83.35
	BQSquare	0.40	1.14	12.08	2.37	57.82	64.58
	BlowingBubbles	0.86	1.15	13.12	2.73	63.42	68.32
	RaceHorses	1.88	1.21	17.29	4.75	78.95	84.95
Media		0.87	1.17	14.62	3.55	70.37	77.09

**Figura 4.13:** Comparación entre el codificador base (barra izquierda) y el codificador con el algoritmo de pre-análisis (barra derecha)

el hecho de que el número de operaciones SAD realizadas en esta parte se reduce un 77.09 %.

Como consecuencia directa de esta reducción de la complejidad computacional del codificador, se da la redistribución de tiempos presente en la Figura 4.13. Estos resultados han sido extraídos a partir de la media de todos los QP codificados. Como puede verse, el tiempo empleado en la estimación de movimiento entera unidireccional (etiqueta “ME IME (Uni-)”) se ve notablemente reducido por las dos causas anteriormente mencionadas. Por su parte, también se ve reducida la parte del módulo ME que incluye la selección de los predictores mediante AMVP (etiqueta “ME Otros”). El resto de módulos permanece inalterado. Cabe añadir también que la figura incluye el tiempo correspondiente al algoritmo de pre-análisis. Sin embargo, éste es tan reducido que no puede apreciarse al ser menor que el 0.5 % del tiempo total.

4.6. Conclusiones

Al principio de este capítulo, se ha descrito la gran necesidad existente de creación de algoritmos rápidos de codificación derivada de la enorme complejidad computacional del estándar HEVC. En este sentido, hemos propuesto la arquitectura de un algoritmo de pre-análisis que tiene como objetivo recopilar información preliminar sobre las secuencias a codificar y calcular estimaciones que permitirán al proceso de codificación reducir su tiempo de ejecución.

Como punto inicial del desarrollo de dicho algoritmo de pre-análisis, este capítulo ha descrito el diseño de una etapa de estimación de movimiento previa a la codificación que permite disponer de una aproximación de los vectores de movimiento calculados por el módulo ME. De esta manera, es posible reducir la complejidad del algoritmo AMVP e igualmente reducir el área de búsqueda de la estimación de movimiento llevada a cabo en la fase de codificación. Tras ello, se ha llevado a cabo la parametrización del algoritmo, llegando a la conclusión de que la configuración óptima para el mismo es emplear un rango de búsqueda de pre-análisis de 64, un tamaño de bloque de pre-análisis de 16×16 y un rango de búsqueda del módulo ME de 4.

Tras una evaluación del algoritmo propuesto empleando la configuración anterior, se obtuvo que es posible reducir más de un 75 % de las operaciones SAD llevadas a cabo en la estimación de movimiento, lo que repercute de forma directa en la reducción de un 70 % del tiempo de la estimación de movimiento entera unidireccional, y de casi un 15 % del tiempo total de codificación. Como consecuencia, la eficiencia de codificación se ve ligeramente afectada en un 0.9 % en términos de BD-rate.

En definitiva, el algoritmo de pre-análisis ha demostrado ser una buena alternativa para acelerar el módulo de ME del codificador. Igualmente, también se muestra como firme candidato para proveer información a otros módulos con el mismo objetivo.

Capítulo 5

ANTEPROYECTO DE TESIS DOCTORAL

Este capítulo cierra el presente documento. En él se expondrán las conclusiones que ha sido posible obtener no sólo a partir del trabajo de investigación previamente expuesto, sino también de los contenidos desarrollados en las asignaturas del Máster Universitario en Tecnologías Informáticas Avanzadas y de los estudios llevados a cabo. A partir de estas conclusiones, se podrá definir la línea de investigación que seguirá la Tesis Doctoral correspondiente, para lo cual será necesario especificar sus objetivos y una planificación estimada de las tareas que la componen.

5.1. Contribuciones a la Tesis Doctoral

A través del estudio del arte desarrollado en el Capítulo 3 y el trabajo de la asignatura Grid Computing, se ha podido llevar a cabo un profundo análisis del estándar HEVC. Mediante ellos, se ha analizado las herramientas y técnicas de codificación más importantes que han sido incluidas en el estándar. Éstas permiten explicar la gran complejidad computacional que alcanzan los codificadores basados en HEVC frente a otros como los de H.264/AVC, incluso bajo las mismas condiciones. De esta manera, es posible justificar la necesidad de trabajar en técnicas de codificación alternativas o complementarias que permitan reducir el tiempo requerido por el proceso de codificación.

Por su parte, los trabajos realizados tanto en la asignatura Metodologías y Técnicas de Investigación en Informática como en Redes Multimedia permitieron llevar a cabo una revisión sistemática de la literatura relacionada con HEVC. En ella se pudo ver que existe un cierto número de trabajos relacionados con algoritmos paralelos de codificación para este estándar. No obstante, muchos de ellos poseen ciertas desventajas

o limitaciones que hacen necesario seguir trabajando de manera complementaria en la parte secuencial del codificador.

Teniendo en cuenta todo lo anterior, una de las contribuciones de este trabajo y, consecuentemente, de la Tesis Doctoral es la de cubrir la necesidad existente de buscar algoritmos de codificación rápidos para reducir la complejidad del codificador basado en HEVC. De hecho, hoy en día apenas existen trabajos que consigan alcanzar la codificación en tiempo real con codificadores basados en este estándar, y los que lo hacen requieren de arquitecturas paralelas de gran potencia de cálculo. Por ejemplo, en [21] utilizan dos computadores interconectados con 32 *cores* cada uno para codificar secuencias 4k en tiempo real.

Por otro lado, si bien los algoritmos de análisis previo a la codificación no son nuevos en sí [36, 37], no existen trabajos relacionados que hagan un análisis detallado de las características que ha de tener este tipo de algoritmos, especialmente en el campo de HEVC, tal y como se ha visto en el listado de obras relacionadas de la Sección 4.2. Por ello, merece la pena destacar también la contribución realizada en el sentido de demostrar cuáles son los parámetros más adecuados para este tipo de algoritmos en este nuevo estándar.

En definitiva, las asignaturas cursadas en el máster y especialmente este trabajo han servido de gran ayuda para establecer la base de lo que será la Tesis Doctoral a desarrollar en los próximos meses.

5.2. Descripción del anteproyecto de Tesis Doctoral

A continuación, se presentará el contexto en que se llevará a cabo la Tesis Doctoral a desarrollar, la cual seguirá la línea del trabajo de investigación descrito en este documento. Tras ello, se presentarán los objetivos a conseguir y las tareas a realizar, así como una planificación temporal y de recursos del proyecto.

5.2.1. Introducción

Como se ha comentado a lo largo del documento, los contenidos multimedia tales como imágenes, vídeo o audio están muy presentes en nuestra diaria a través de tecnologías y dispositivos cada vez más avanzados, y en ámbitos tan diversos como la telemedicina, la videoconferencia o la industria cinematográfica. Cada día se producen y transmiten millones de vídeos por las redes de comunicaciones como Internet. Todo ello se debe al continuo avance de la tecnología y al cambio en los patrones de consumo de los usuarios, los cuales desean disponer de esos contenidos en cualquier momento y lugar. Sin embargo, un contenido de vídeo sin comprimir puede consumir demasiados

recursos, tanto para su transmisión como para su almacenamiento. Es por ello por lo que la compresión de vídeo se vuelve indispensable. Además, los usuarios demandan contenidos con calidades y resoluciones cada vez mayores, lo que incita el desarrollo de estándares de compresión cada vez más eficientes, capaces de alcanzar grandes *ratios* de compresión, preservando la calidad visual.

En los últimos años, H.264/AVC [5] ha sido el estándar de codificación de vídeo más ampliamente utilizado por la industria audiovisual gracias a sus capacidades de eficiencia de codificación. Tanto es así que la mayoría de cámaras de vídeo profesionales y domésticas usan en la actualidad este estándar, e incluso es el formato empleado en los discos Blu-ray. No obstante, las recientes demandas de soporte para la codificación de vídeo de grandes resoluciones como 4k y 8k, así como la necesidad de disponer de un nuevo codificador que mejorase sustancialmente las prestaciones de H.264/AVC, motivaron el nacimiento del estándar HEVC [6].

El precio a pagar por la mayor eficiencia de codificación de este nuevo estándar es, sin embargo, el elevadísimo coste computacional que conlleva [10]. Por ello, hay una necesidad evidente de acelerar este proceso de codificación. Una de las opciones más recurridas para ello es la computación paralela [16], basada en la idea de que un trabajo puede dividirse en pequeñas tareas que se pueden ejecutar de forma simultánea. No obstante, esta opción presenta limitaciones que hacen que sea necesario seguir esforzándose en encontrar algoritmos rápidos de codificación. Así, numerosos trabajos se esfuerzan en acelerar los diversos módulos que forman parte del codificador [27], especialmente el de estimación de movimiento [31, 32, 33]. Pese a todo, todavía no se ha conseguido llegar a la barrera establecida por la codificación en tiempo real de contenidos de alta calidad en HEVC con un uso de recursos asequible, lo que implica disponer de un estándar con grandes características, pero difícil aplicabilidad en la actualidad.

5.2.2. Objetivos

Tal y como se puede extraer a partir de los anteriores apartados, el objetivo de la Tesis Doctoral a desarrollar es reducir la complejidad computacional del codificador del estándar HEVC mediante algoritmos de pre-análisis que den soporte a los diferentes módulos que lo componen, manteniendo en cualquier caso la eficiencia de codificación que éste provee. Este objetivo no sólo engloba el hecho de desarrollar dicho algoritmo, sino también justificar cada una de las decisiones de diseño llevadas a cabo y comparar éstas con otras alternativas.

Teniendo en cuenta el objetivo principal de la Tesis Doctoral, existe una serie de subobjetivos relacionados que se enumeran a continuación:

- *Realizar un análisis exhaustivo del estándar HEVC y su codificador.*

Si bien en este trabajo se ha presentado una descripción del estándar, éste contiene un gran nivel de detalle que es necesario conocer y analizar para poder extraer el máximo provecho de los algoritmos de pre-análisis desarrollados. Posteriormente, es necesario conocer también el modo de operación de los distintos módulos que componen un codificador típico de HEVC con el objetivo de determinar de qué manera un algoritmo de pre-análisis podría proveerles soporte.

- *Dar soporte a la predicción de tipo inter.*

Este objetivo comprende todas las tareas enfocadas a acelerar el proceso de estimación de movimiento y de búsqueda de vectores de movimiento. En él se incluye el trabajo de investigación desarrollado en este documento que, si bien es una propuesta íntegra de por sí, todavía puede ser ampliado para, por ejemplo, contemplar el uso inteligente de los *frames* de referencia en función de la información extraída en la fase de pre-análisis.

- *Dar soporte a la predicción de tipo intra.*

De manera análoga al objetivo anterior, en éste se trata de acelerar el proceso de interpolación llevado a cabo en la predicción de tipo intra. Para ello, la fase de pre-análisis puede determinar de manera aproximada el modo o la dirección, según el caso, que el filtro debe adoptar en este tipo de predicción.

- *Dar soporte a la decisión de modo y al particionamiento CTU.*

A partir de los algoritmos desarrollados a raíz de los objetivos anteriores, sería posible obtener una estimación de los costes que supondría codificar un determinado PU empleando predicción inter o intra. Con estos costes, por tanto, se podría decidir el modo de predicción en la propia etapa de pre-análisis. A su vez, estos modos y la información de predicción asociada podrían utilizarse también para determinar el posible tamaño de CU a emplear, limitando el tamaño del árbol de particionamiento CTU recorrido en el codificador.

- *Integrar la propuesta en arquitecturas heterogéneas.*

En el estado en que se encuentra el trabajo de investigación realizado, el algoritmo de pre-análisis supone una carga mínima dentro del codificador. No obstante, se prevé que esta situación cambie conforme vaya incluyendo la carga detallada en objetivos anteriores. En tal caso, se hará uso de arquitecturas heterogéneas basadas en GPU u otro tipo de dispositivos con el objetivo de reducir la complejidad computacional de este algoritmo.

5.2.3. Tareas a realizar

En alineación con los objetivos, es posible definir una serie de tareas a realizar a lo largo de la Tesis Doctoral. Estas tareas tienen un carácter general y son divididas a su

vez en subtareas, las cuales tienen una duración determinada. Hay que tener en cuenta, no obstante, que algunas de estas tareas han podido ser parcial o totalmente cubiertas en el desarrollo de este u otros trabajos.

■ *Análisis del estado del arte y trabajos relacionados.*

Esta tarea se encuentra enfocada al estudio de la materia relacionada con la Tesis Doctoral a desarrollar, que en este caso se corresponde con el estándar HEVC. Si bien ya ha comenzado a llevarse a cabo en buena medida gracias a este Trabajo Fin de Máster, es cierto que es necesario profundizar en diversos aspectos. Por ello, será una tarea recurrente previa al desarrollo de las distintas partes que componen el algoritmo de pre-análisis. Añadiremos también en este punto el análisis del estado de arte relacionado con las arquitecturas heterogéneas, puesto que será necesario para integrar el algoritmo en este tipo de plataformas. Así, algunas subtareas a contemplar en este apartado son:

- Análisis general del estándar HEVC.
- Estudio de los trabajos relacionados en la temática sobre la aceleración del *encoder* y los algoritmos de pre-análisis.
- Análisis del módulo de predicción tipo inter.
- Análisis del módulo de predicción tipo intra.
- Estudio de algoritmos de decisión de modo y particionamiento CTU.
- Estudio de las diferentes alternativas de arquitecturas heterogéneas.

■ *Investigación y desarrollo.*

Esta tarea conforma la parte central de la Tesis Doctoral de la cual dependen todas las demás. Incluye la implementación del algoritmo de pre-análisis en todas sus fases, así como su evaluación frente al codificador base y otras alternativas:

- Desarrollo del algoritmo de soporte a la estimación de movimiento entera unidireccional.
- Desarrollo del algoritmo de soporte a la elección de *frames* de referencia.
- Desarrollo del algoritmo de soporte a la predicción de tipo intra.
- Desarrollo del algoritmo de decisión de modo.
- Desarrollo del algoritmo de particionamiento CTU.
- Integración del algoritmo en arquitecturas heterogéneas.

Nótese que mientras las tres primeras subtareas pueden formar parte de propuestas individuales e independientes entre sí, el resto necesita que las anteriores hayan sido implementadas para su desarrollo. Por tanto, con el objetivo de mantener una propuesta íntegra y compacta, estas tareas se llevarán a cabo de manera ordenada e incremental.

■ *Formación.*

En todo el proceso correspondiente al desarrollo de la Tesis Doctoral y, en realidad, a lo largo de toda la vida de un investigador, resulta esencial llevar a cabo tareas de formación que ayuden a incrementar el conocimiento de éste en su área de especialización. Algunas subtareas relacionadas con este apartado que podemos incluir en esta planificación son:

- Cursar el Máster Universitario en Tecnologías Informáticas Avanzadas.
- Realizar estancias en otras universidades o centros de investigación.
- Formarse en los lenguajes empleados para controlar los dispositivos de las arquitecturas heterogéneas, como por ejemplo OpenCL en el caso de las GPUs.

■ *Generación de documentos.*

Los progresos alcanzados durante la Tesis Doctoral han de recogerse en documentos que lo reflejen y que permitan determinar si los resultados obtenidos son los esperados o es necesario realizar alguna modificación en el planteamiento de la misma. En este sentido, las subtareas recogidas en este apartado son:

- Elaborar la memoria del Trabajo Fin de Máster (este documento).
- Elaboración de las memorias anuales de seguimiento.
- Redacción y revisión de la memoria de Tesis Doctoral.

Tras este listado de tareas, cabe añadir que también pueden añadirse aquéllas relacionadas con la divulgación del trabajo científico a través de artículos y congresos. No obstante, dicha tarea es un trabajo que se realiza a lo largo de toda la carrera investigadora. En nuestro caso, se tratará de que tras la consecución de cada una de las tareas de desarrollo, se reflejen los resultados en forma de una o varias publicaciones científicas.

5.2.4. Cronograma temporal

Las tareas y subtareas identificadas en la sección anterior pueden ser organizadas de manera gráfica en un diagrama de Gantt, de forma que es posible establecer un orden temporal de los eventos. Para elaborarlo, tomaremos como fecha inicial no la de finalización del Máster Universitario en Tecnologías Informáticas Avanzadas, sino aquélla que se sitúa poco antes de iniciar la estancia en la *Technische Universität Berlin* en septiembre de 2014, que es cuando se comenzó a trabajar en esta línea de investigación.

Teniendo lo anterior en cuenta, la Figura 5.1 muestra un diagrama aproximado de la planificación a seguir en esta Tesis Doctoral. Las barras de progreso se muestran

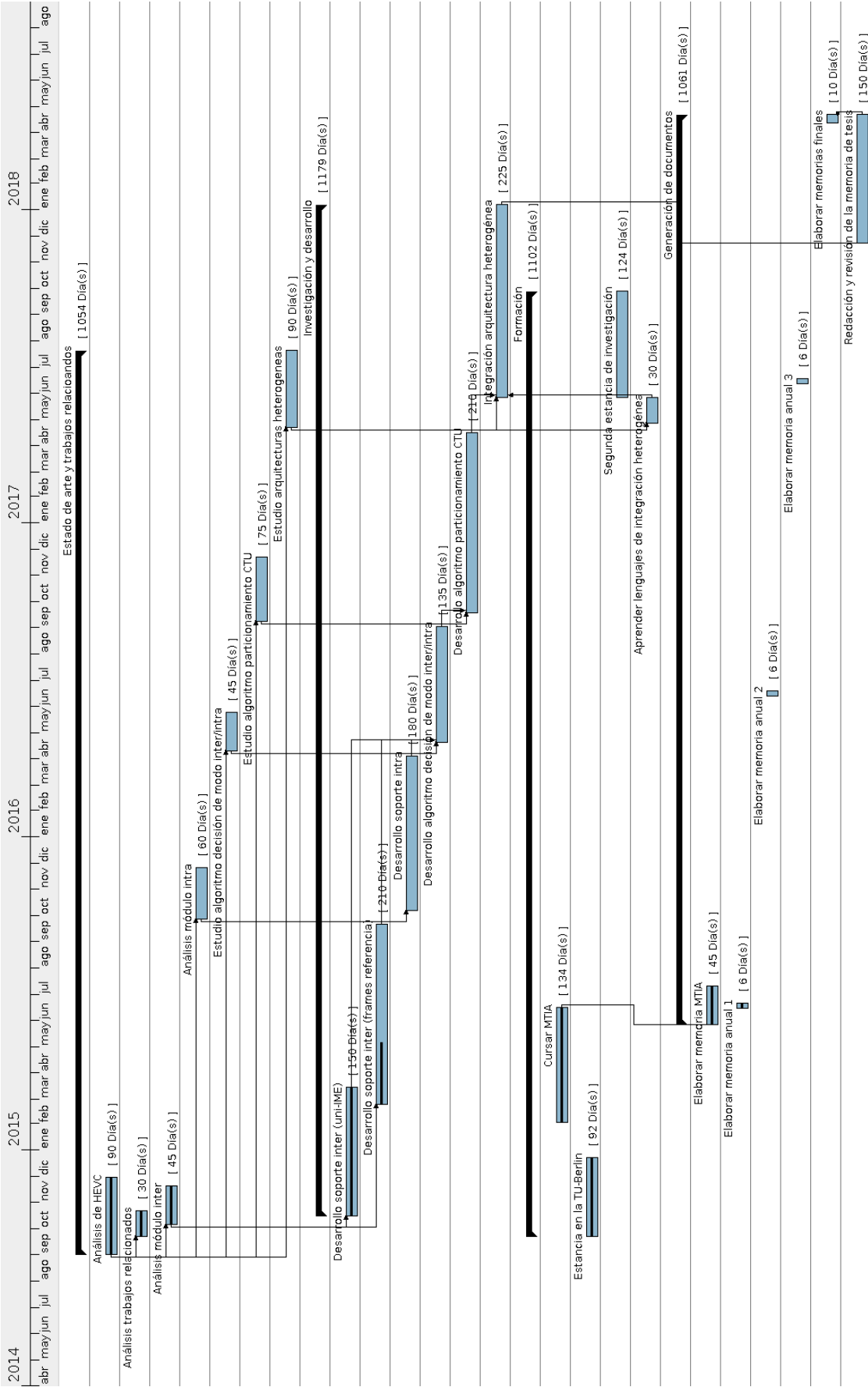


Figura 5.1: Planificación temporal de la Tesis Doctoral propuesta

rellenas en función de su completitud hasta el momento. Nótese, además, que la tarea de análisis de HEVC tiene un tamaño relativamente reducido. Esto se debe a que ya se han llevado a cabo estudios del estándar en anteriores trabajos de investigación y no es necesario dedicarle tiempo adicional.

5.2.5. Recursos necesarios

La Tesis Doctoral se desarrollará en el grupo de *Redes y Arquitecturas de Altas Prestaciones* (RAAP) del Instituto de Investigación en Informática de Albacete (I3A). Este grupo cuenta con una dilatada y reconocida experiencia no sólo en las temáticas que su propio nombre indica, sino también en el ámbito de la codificación de vídeo. Además, forma parte de diversos proyectos de investigación de carácter nacional e internacional, lo que asegura la financiación de los recursos que sus miembros requieren.

En lo relativo al material necesario para llevar a cabo la Tesis Doctoral, el grupo cuenta con la infraestructura *hardware* suficiente para su ejecución. Se dispone de *clusters* de computación que permiten albergar múltiples procesos de codificación al mismo tiempo, lo que supone una gran reducción del tiempo necesario para realizar las pruebas correspondientes. Adicionalmente, se dispone de financiación procedente de proyectos de infraestructura que permitirán adquirir nuevos nodos de computación, algunos de ellos equipados con GPUs y una tarjeta Intel Xeon Phi [12]. Este hecho permitirá trasladar el algoritmo de pre-análisis a algunas de las plataformas y tecnologías más novedosas de la alta computación.

BIBLIOGRAFÍA

- [1] “Cisco Visual Networking Index: Forecast and Methodology, 2014-2019,” Cisco, May 2015.
- [2] *Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s. ISO/IEC Recommendation 11172-2*, ISO/IEC, Aug. 1993.
- [3] *Generic coding of moving pictures and associated audio information: Video. ISO/IEC Recommendation 13818-2*, ISO/IEC, May 1996.
- [4] *Generic Coding of Audio-Visual Objects - Part 2: Visual. ISO/IEC Recommendation 14496-2*, ISO/IEC, Dec. 1999.
- [5] *Advanced Video Coding for Generic Audiovisual Services. ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, ISO/IEC and ITU-T, May 2003.
- [6] *High Efficiency Video Coding. ITU-T Recommendation H.265 and ISO/IEC 23008-2 (HEVC)*, ISO/IEC and ITU-T, Apr. 2013.
- [7] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC),” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.
- [8] H.264/AVC Reference Software. Fraunhofer HHI. Accessed: Jul. 2015. [Online]. Available: <http://iphome.hhi.de/suehring/tml/>
- [9] HEVC Test Model (HM) Reference Software. Fraunhofer HHI. Accessed: Jun. 2015. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/
- [10] F. Bossen, B. Bross, K. Sühning, and D. Flynn, “HEVC Complexity and Implementation Analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.
- [11] Embedded Systems Architecture (AES) homepage. Technische Universität Berlin. Accessed: Jul. 2015. [Online]. Available: <http://www.aes.tu-berlin.de/>

- [12] J. Fang, H. Sips, L. Zhang, C. Xu, Y. Che, and A. L. Varbanescu, “Test-driving Intel Xeon Phi,” in *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering*. New York, USA: ACM, Mar. 2014, pp. 137–148.
- [13] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. Wiley, Apr. 2010.
- [14] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [15] D. P. Mehta and S. Sahni, Eds., *Handbook of Data Structures and Applications*. Chapman & Hall/CRC, Oct. 2004.
- [16] C. C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, “Parallel Scalability and Efficiency of HEVC Parallelization Approaches,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1827–1838, Dec. 2012.
- [17] K. Chen, J. Sun, Y. Duan, and Z. Guo, “A Novel Wavefront-Based High Parallel Solution for HEVC,” *IEEE Trans. Circuits Syst. Video Technol.*, Mar. 2015, Online First.
- [18] P. Piñol, H. Migallón, O. López-Granado, and M. P. Malumbres, “Slice-based Parallel Approach for HEVC Encoder,” *J. Supercomput.*, vol. 71, no. 5, pp. 1882–1892, May 2015.
- [19] A. J. Díaz-Honrubia, G. Cebrián-Márquez, J. L. Martínez, P. Cuenca, J. M. Puerta, and J. A. Gámez, “Low-Complexity Heterogeneous Architecture for H.264/HEVC Video Transcoding,” *J. Real-Time Image Proc.*, Dec. 2014, Online First.
- [20] P. Piñol, H. Migallón, O. López-Granado, and M. P. Malumbres, “Parallel Strategies Analysis over the HEVC Encoder,” *J. Supercomput.*, vol. 70, no. 2, pp. 671–683, Nov. 2014.
- [21] T. K. Heng, W. Asano, T. Itoh, A. Tanizawa, J. Yamaguchi, T. Matsuo, and T. Kodama, “A highly parallelized H.265/HEVC real-time UHD software encoder,” in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 1213–1217.
- [22] M. Kim, D. Kim, K. Kim, and W. W. Ro, “Exploiting Thread-Level Parallelism on HEVC by Employing Reference Dependency Graph,” *IEEE Trans. Circuits Syst. Video Technol.*, Mar. 2015, Online First.

- [23] C. Yan, Y. Zhang, J. Xu, F. Dai, L. Li, Q. Dai, and F. Wu, “A Highly Parallel Framework for HEVC Coding Unit Partitioning Tree Decision on Many-core Processors,” *IEEE Signal Process. Lett.*, vol. 21, no. 5, pp. 573–576, May 2014.
- [24] J. Zhang, F. Dai, Y. Ma, and Y. Zhang, “Highly parallel mode decision method for HEVC,” in *Picture Coding Symposium (PCS)*, Dec. 2013, pp. 281–284.
- [25] K. Miyazawa, H. Sakate, S.-i. Sekiguchi, N. Motoyama, Y. Sugito, K. Iguchi, A. Ichigaya, and S.-i. Sakaida, “Real-time hardware implementation of HEVC video encoder for 1080p HD video,” in *Picture Coding Symposium (PCS)*, Dec. 2013, pp. 225–228.
- [26] G. Cebrián-Márquez, J. L. Hernández-Losada, J. L. Martínez, P. Cuenca, M. Tang, and J. Wen, “Accelerating HEVC using Heterogeneous Platforms,” *J. Supercomput.*, vol. 71, no. 2, pp. 613–628, Feb. 2015.
- [27] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, “An Effective CU Size Decision Method for HEVC Encoders,” *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 465–470, Feb. 2013.
- [28] S. Ahn, B. Lee, and M. Kim, “A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 422–435, Mar. 2015.
- [29] G. Van Wallendael, S. Van Leuven, J. De Cock, P. Lambert, R. Van de Walle, J. Barbarien, and A. Munteanu, “Improved intra mode signaling for HEVC,” in *IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2011, pp. 1–6.
- [30] Y. Liu, X. Liu, and B. Shao, “Fast Intra-mode Decision Algorithm for HEVC Based on Texture Direction,” in *IEEE 17th International Conference on Computational Science and Engineering (CSE)*, Dec. 2014, pp. 1047–1050.
- [31] L. Shen, Z. Zhang, and Z. Liu, “Adaptive Inter-Mode Decision for HEVC Jointly Utilizing Inter-Level and Spatiotemporal Correlations,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1709–1722, Oct. 2014.
- [32] J. Lee, S. Kim, K. Lim, and S. Lee, “A Fast CU Size Decision Algorithm for HEVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 411–421, Jul. 2014.
- [33] J. Xiong, H. Li, Q. Wu, and F. Meng, “A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence,” *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.

- [34] C. E. Rhee, K. Lee, T. S. Kim, and H.-J. Lee, “A survey of fast mode decision algorithms for inter-prediction and their applications to high efficiency video coding,” *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1375–1383, Nov. 2012.
- [35] x264 source repository. VideoLAN. Accessed: Jul. 2015. [Online]. Available: <http://git.videolan.org/>
- [36] J. Garrett-Glaser, “A Novel Macrobblock-Tree Algorithm for High-Performance Optimization of Dependent Video Coding in H.264/AVC,” Department of Computer Science. Harvey Mudd College, Tech. Rep., 2009.
- [37] L. Sun, O. C. Au, C. Zhao, and F. H. Huang, “Rate distortion modeling and adaptive rate control scheme for high efficiency video coding (HEVC),” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Jun. 2014, pp. 1933–1936.
- [38] T. Mallikarachchi, A. Fernando, and H. K. Arachchi, “Efficient coding unit size selection based on texture analysis for HEVC intra prediction,” in *IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2014, pp. 1–6.
- [39] G. Tian and S. Goto, “Content adaptive prediction unit size decision algorithm for HEVC intra coding,” in *Picture Coding Symposium (PCS)*, May 2012, pp. 405–408.
- [40] N. Purnachand, L. N. Alves, and A. Navarro, “Improvements to TZ search motion estimation algorithm for multiview video coding,” in *19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Apr. 2012, pp. 388–391.
- [41] G. Bjøntegaard, “Calculation of average PSNR differences between RD-curves,” ITU-T Video Coding Experts Group (VCEG), Tech. Rep. VCEG-M33, 2001.
- [42] F. Bossen, “Common test conditions and software reference configurations,” Joint Collaborative Team on Video Coding (JCT-VC), Tech. Rep. JCTVC-L1100, Jan. 2013.
- [43] J. Charles, P. Jassi, A. Narayan S., A. Sadat, and A. Fedorova, “Evaluation of the Intel Core i7 Turbo Boost feature,” in *IEEE International Symposium on Workload Characterization*, Oct. 2009, pp. 188–197.
- [44] D. Koufaty and D. T. Marr, “Hyperthreading technology in the netburst microarchitecture,” *IEEE Micro*, vol. 23, no. 2, pp. 56–65, Mar. 2003.

Apéndice A

CURRICULUM VITAE

DATOS PERSONALES

APELLIDOS, NOMBRE: Cebrián Márquez, Gabriel
DNI: 47.096.875-M
NACIONALIDAD: Española
FECHA DE NACIMIENTO: 5 de julio de 1991
TELÉFONO: (+34) 967 599 200 Ext. 2651
CORREO ELECTRÓNICO: Gabriel.Cebrian@uclm.es

FORMACIÓN ACADÉMICA

2014 – ACTUALMENTE ***Doctorado en Tecnologías Informáticas Avanzadas (R.D. 1393/2007)***
Universidad de Castilla-La Mancha

2013 – ACTUALMENTE ***Máster Universitario en Tecnologías Informáticas Avanzadas***
Universidad de Castilla-La Mancha

2013 – 2014 ***Máster Universitario en Ingeniería Informática***
Universidad de Castilla-La Mancha
Premio Fin de Máster

2009 – 2013 ***Grado en Ingeniería Informática***
Universidad de Castilla-La Mancha
Intensificación en Ingeniería de Computadores
Premio Fin de Grado

FORMACIÓN COMPLEMENTARIA

06/2013 ***Título oficial B2 de Inglés***
Escuela Oficial de Idiomas de Albacete

EXPERIENCIA LABORAL Y ACADÉMICA

15/09/2014 –	Becario de investigación FPU
ACTUALMENTE	Redes y Arquitecturas de Altas Prestaciones (RAAP) Instituto de Investigación en Informática de Albacete (I3A)
01/01/2014 –	Contratado con cargo a proyectos de investigación
14/09/2014	Redes y Arquitecturas de Altas Prestaciones (RAAP) Instituto de Investigación en Informática de Albacete (I3A)
20/09/2013 –	Becario con cargo a proyectos de investigación
31/12/2013	Redes y Arquitecturas de Altas Prestaciones (RAAP) Instituto de Investigación en Informática de Albacete (I3A)
22/02/2013 –	Becario de colaboración
31/08/2013	Redes y Arquitecturas de Altas Prestaciones (RAAP) Instituto de Investigación en Informática de Albacete (I3A)
12/07/2012 –	Desarrollador (becario)
12/10/2012	Area Project Solutions

PUBLICACIONES

REVISTAS

A. J. Díaz-Honrubia, G. Cebrián-Márquez, J. L. Martínez, P. Cuenca, J. M. Puerta, and J. A. Gámez, “Low-Complexity Heterogeneous Architecture for H.264/HEVC Video Transcoding,” *J. Real-Time Image Proc.*, Dec. 2014, Online First.

Factor de impacto JCR 2014: 2.020 (Q1)

G. Cebrián-Márquez, J. L. Hernández-Losada, J. L. Martínez, P. Cuenca, M. Tang, and J. Wen, “Accelerating HEVC using Heterogeneous Platforms,” *J. Supercomput.*, vol. 71, no. 2, pp. 613–628, Feb. 2015.

Factor de impacto JCR 2014: 0.858 (Q3)

CONGRESOS

G. Cebrián-Márquez, C. C. Chi, J. L. Martínez, P. Cuenca, M. Álvarez-Mesa, S. Sanz-Rodríguez, and B. Juurlink, “Reducing HEVC Encoding Complexity Using Two-Stage Motion Estimation,” in *Visual Communications and Image Processing Conference*, Dec. 2015, pp. 13–16, Submitted for review.

G. Cebrián-Márquez, J. L. Martínez, and P. Cuenca, “Limitando el uso de cuadros de referencia en HEVC,” in *Actas de las XXVI Jornadas de Paralelismo*. Sociedad de Arquitectura y Tecnología de Computadores (SARTECO), Sep. 2015, Accepted.

H. Migallón, J. L. Hernández-Losada, G. Cebrián-Márquez, P. Piñol, J. L. Martínez, O. López-Granado, and M. P. Malumbres, “OpenMP HEVC Parallel Version based on a GOP Approach,” in *Proceedings of the Ninth International Conference on Engineering Computational Technology*, P. Iványi and B. H. Topping, Eds., Sep. 2014.

G. Cebrián-Márquez, J. L. Martínez, P. Cuenca, M. Tang, and J. Wen, “Accelerating HEVC using GPU-based heterogeneous platforms,” in *Proceedings of the 14th International Conference on Mathematical Methods in Science and Engineering*, J. V. Aguiar, Ed., vol. 1, Jul. 2014, pp. 288–299.

PROYECTOS ---

TIN2012-38341-C04-04, Mejora de la Arquitectura de Servidores, Servicios y Aplicaciones. Ministerio de Economía y Competitividad. Francisco José Quiles Flor. (Universidad de Castilla-La Mancha). 01/01/2013 – 31/12/2015.

UNCM13-1E-2294, Transcodificador de vídeo heterogéneo H.264/HEVC. Ministerio de Economía y Competitividad. José Luis Martínez Martínez. (Universidad de Castilla-La Mancha). 01/01/2013 – 31/12/2015.

TIN2014-53522-REDT, Red de computación de altas prestaciones sobre arquitecturas paralelas heterogéneas (CAPAP-H5). Ministerio de Ciencia e Innovación. (22 grupos de investigación de diversas universidades españolas). 15/06/2014 – 14/06/2015.

CONTRATOS ---

Transcodificador de vídeo 2D/3D heterogéneo H.264/HEVC. AICOX Soluciones S.A.; Servicios Audiovisuales OVERON. Pedro Ángel Cuenca Castillo. 16/07/2012 – 16/07/2014.

ESTANCIAS EN CENTROS EXTRANJEROS ---

Embedded Systems Architecture (AES). Technische Universität Berlin (TU-Berlin). Berlin, Alemania. 22/09/2014 – 22/12/2014. Look-Ahead Analysis for the HEVC Encoder.